Digital curvelet transform: strategy, implementation, and experiments

David L. Donoho, Mark R. Duncan

David L. Donoho, Mark R. Duncan, "Digital curvelet transform: strategy, implementation, and experiments," Proc. SPIE 4056, Wavelet Applications VII, (5 April 2000); doi: 10.1117/12.381679

Event: AeroSense 2000, 2000, Orlando, FL, United States

Digital Curvelet Transform: Strategy, Implementation and Experiments

David L. Donoho & Mark R. Duncan
Department of Statistics
Stanford University

November, 1999

Abstract

Recently, Candès and Donoho (1999) introduced the curvelet transform, a new multiscale representation suited for objects which are smooth away from discontinuities across curves. Their proposal was intended for functions f defined on the continuum plane $\mathbf{R}^2$.

In this paper, we consider the problem of realizing this transform for digital data. We describe a strategy for computing a digital curvelet transform, we describe a software environment, `Curvelet256`, implementing this strategy in the case of 256×256 images, and we describe some experiments we have conducted using it. Examples are available for viewing by web browser.

Dedication. To the Memory of Dr. Revira Singer, z"l.

Keywords. Wavelets, Curvelets, Ridgelets, Digital Ridgelet Transform.

Acknowledgements. We would like to thank Emmanuel Candès and Xiaoming Huo for many constructive suggestions, for editorial comments, and lengthy discussions. A reproduction of the Picasso engraving was kindly provided by Ruth Kozodoy, Metropolitan Museum of Art, New York.

This research was supported by NSF grant DMS 95-05152, by NSF grant DMS 98-72890, and by DARPA BAA 98-04.

1 Introduction

1.1 The Importance of being Digital

About fifteen years ago, mathematicians and physicists in France were working intensely on what they called the wavelet transform, a representation of functions defined on the real line $f(t)$, $t \in \mathbf{R}$. Arising from questions in geophysics and mathematical physics, the new transform was conceptually intriguing – offering the possibility of decomposing phenomena naturally into components at multiple scales, with all the interesting possibilities that would entail. Much of the early work was of a theoretical or natural philosophy bent, and one of the fascinating achievements was the exposure of parallels and contrasts with intellectual developments in a wide range of fields in science and technology, from computer-aided design to group representation theory [16].

Today, rather than a topic for intellectual stimulation and development, the wavelet transform is a very practical everyday tool, applied routinely by thousands of researchers in all branches of science and engineering. It could be considered a 'household word' in the halls of scientific and engineering institutions.

One of the key steps in this rather dramatic morphing from concept to practice was the realization that, although the transform was originally considered in a rarified atmosphere – a representation for continuous functions on the continuum line – it had a natural translation into the world of digital signals. This opened the way for a transformation of the topic from elitist to populist.

The story is well-known; see for example Meyer's book [15]. Mallat and Meyer (1986) introduced the concept of *multiresolution analysis* which showed that a certain wavelet transform could be organized as a ladder of component stages, each one involving simply applying some digital filters to certain discrete-time 'signals'. Daubechies (1987) showed that there was a family of short, finite length filters obeying all the constraints imposed by the multiresolution analysis, leading immediately to a fast orthogonal transform of finite-length sequences. Once this discrete wavelet transform was available, scientists and engineers in many fields had a fast and flexible tool for exploring the multiscale structure of their digital data.

1.2 Curvelet Transform

Recently, Candès and Donoho [7] developed a new multiscale transform which they called the *curvelet transform.* Motivated by the needs of image analysis, it was nevertheless first proposed in the context of objects $f(x_1, x_2)$ defined on the continuum plane $(x_1, x_2) \in \mathbf{R}^2$. The transform was designed to represent edges and other singularities along curves much more efficiently than traditional transforms, i.e. using many fewer coefficients for a given accuracy of reconstruction. Roughly speaking, to represent an edge to squared error $1/N$ requires $1/N$ wavelets and only about $1/\sqrt{N}$ curvelets.

The curvelet transform, like the wavelet transform, is a multiscale transform, with frame elements indexed by scale and location parameters. Unlike the wavelet transform, it has directional parameters, and the curvelet pyramid contains elements with a very high degree of directional specificity. In addition, the curvelet transform is based on a certain *anisotropic scaling* principle which is quite different from the *isotropic scaling* of wavelets. The elements obey a special scaling law, where the length of the support of a frame elements and the width of the support are linked by the relation $width \approx length^2$.

All of these properties are very stimulating and have already lead to a range of interesting idealized applications – for example in tomography and in scientific computation [9, 10]. In effect, an understanding of the curvelet transform concept opens one's eyes to the fact that in two and higher dimensions, new multiscale representations are possible, having properties unavailable by wavelets and having stimulating structural features.

While it is possible that this new idea will be quickly forgotten with the passage of time, we feel that the very novel features of the transform - anisotropy, anisotropy scaling - compel further investigation for the moment.

1.3 Digital Curvelet Transform?

In modern life, utilitarianism is king. Long before all the intellectual exploration of curvelets has run its course, the need to explore practical applications will intrude, leading directly to the question *how can we take a curvelet transform on digital data*? For example, one could imagine, based on the idealized applications already reported, that such a digital transform could be valuable in a variety of areas where edges arise in 2-d data – such as image processing, medical imaging, and remote sensing.

Speaking soberly, the curvelet transform definition (at least at the moment) is much more involved than the wavelet transform, and it seems highly unlikely that such a specialized transform could ever enjoy the same kind of widespread audience as the wavelet transform. However, it also seems that the transform is intrinsically interesting because of its structural differences from other existing transforms. The popularity of the wavelet transform ensures there will be substantial, if not overwhelming, interest for any new transform with both substantial similarities and contrasts to wavelets.

In this article, we report a strategy for developing a digital curvelet transform (DCvT), an implementation for 256 by 256 images, and we point the reader to results of first experiments on image data. Our experiments show clearly that with very few digital curvelet terms, one obtains a reconstruction which is surprsingly faithful to the geometry of the edges of the image.

In our view, the specific tools we develop are not conclusive; an authoritative realization of the DCvT remains to be developed. It may be worth keeping in mind that, after the continuous wavelet transform was first defined, it took years of very hard effort by very clever and zealous researchers for the digital wavelet transform to emerge as a coherent, definite tool available for widespread application. In that context, we claim merely that the present effort may inspire further work and may form a useful base for future research.

1.4 Contents

The contents of the article are as follows. In Section 2, we review the curvelet transform for continuous objects defined in $\mathbf{R}^2$. In Section 3, we describe our implementation strategy, which mimics the continuum viewpoint faithfully, and we describe the required components of our transform, such as the digital ridgelet transform. In Section 4, we describe the software library we have created, and some of the tasks it is able to perform. In Section 5, we give an inventory of some of the experiments we have performed. Section 6 discusses some directions for further work.

2 Curvelet Transform

The curvelet tight frame for $L^2(\mathbf{R}^2)$ is a collection of analyzing elements $\gamma_\mu = \gamma_\mu(x_1, x_2)$ indexed by tuples $\mu \in M'$ to be described below. It has been defined in [8, 7] and has the following key properties:

- Transform Definition:

$$\alpha_\mu \equiv \langle f, \gamma_\mu \rangle, \qquad \mu \in M'.$$

- Parseval Relation:
$$\|f\|_2^2 = \sum_{\mu \in M'} |\alpha_\mu|^2.$$
- L^2 Reconstruction Formula:
$$f = \sum_{\mu \in M'} \langle f, \gamma_\mu \rangle \gamma_\mu.$$

These formal properties are very similar to those one expects from an orthonormal basis, and reflect an underlying stability of representation.

2.1 Analysis

There is a procedural definition of the transform.

- *Subband Decomposition.* We define a bank of subband filters P_0, $(\Delta_s, s \geq 0)$. The object f is filtered into subbands:
$$f \mapsto (P_0 f, \Delta_1 f, \Delta_2 f, \ldots).$$
The different subbands $\Delta_s f$ contain details about 2^{-2s} wide.

- *Smooth Partitioning.* We define a collection of smooth windows $w_Q(x_1, x_2)$ localized around dyadic squares
$$Q = [k_1/2^s, (k_1+1)/2^s) \times [k_2/2^s, (k_2+1)/2^s)$$
Multiplying a function by the corresponding window function w_Q produces a result localized near Q. Doing this for all Q at a certain scale, i.e. for all $Q = Q(s, k_1, k_2)$ with k_1 and k_2 varying but s fixed, produces a smooth dissection of the function into 'squares'. In this stage of the procedure, we apply this windowing dissection to each of the subbands isolated in the previous stage of the algorithm.
$$\Delta_s f \mapsto (w_Q \Delta_s f)_{Q \in \mathcal{Q}_s}.$$

- *Renormalization.* For a dyadic square Q, let
$$(T_Q f)(x_1, x_2) = 2^s f(2^s x_1 - k_1, 2^s x_2 - k_2)$$
denote the operator which transports and renormalizes f so that the part of the input supported near Q becomes the part of the output supported near $[0,1]^2$.

  In this stage of the procedure, each 'square' resulting in the previous stage is renormalized to unit scale
$$g_Q = (T_Q)^{-1}(w_Q \Delta_s f), \qquad Q \in \mathcal{Q}_s.$$

- *Ridgelet Analysis.* Each 'square' is analyzed in the orthonormal ridgelet system. This is a system of basis elements ρ_λ making an orthobasis for $L^2(\mathbf{R}^2)$:
$$\alpha_\mu = \langle g_Q, \rho_\lambda \rangle, \qquad \mu = (Q, \lambda).$$

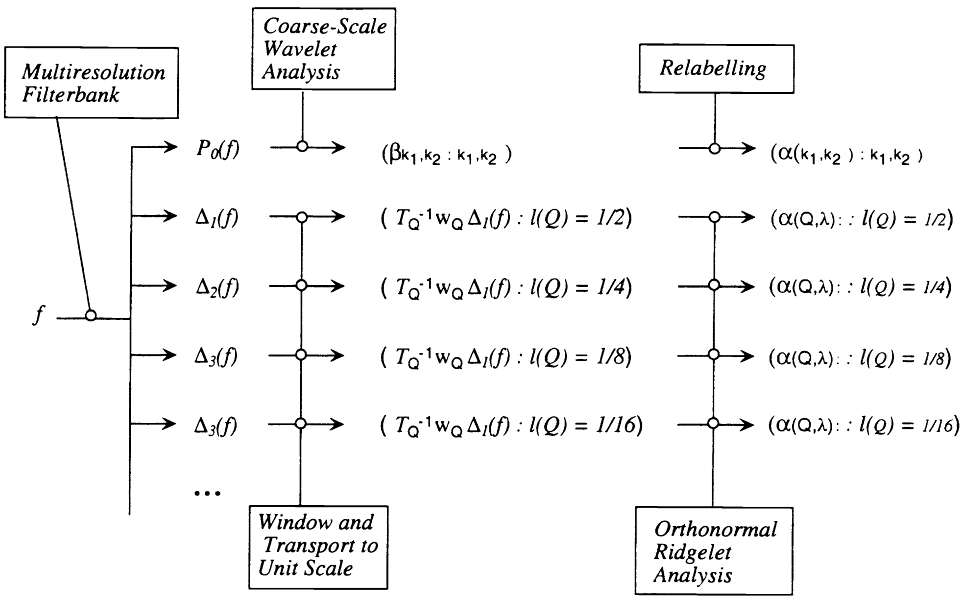
Multiresolution Filterbank
Coarse-Scale Wavelet Analysis
Relabelling
f
$P_0(f)$
$\Delta_1(f)$
$\Delta_2(f)$
$\Delta_3(f)$
$\Delta_3(f)$
...
$(\beta_{k_1,k_2} : k_1,k_2)$
$(T_Q^{-1} w_Q \Delta_l(f) : l(Q) = 1/2)$
$(T_Q^{-1} w_Q \Delta_l(f) : l(Q) = 1/4)$
$(T_Q^{-1} w_Q \Delta_l(f) : l(Q) = 1/8)$
$(T_Q^{-1} w_Q \Delta_l(f) : l(Q) = 1/16)$
$(\alpha(k_1,k_2) : k_1,k_2)$
$(\alpha(Q,\lambda) : : l(Q) = 1/2)$
$(\alpha(Q,\lambda) : : l(Q) = 1/4)$
$(\alpha(Q,\lambda) : : l(Q) = 1/8)$
$(\alpha(Q,\lambda) : : l(Q) = 1/16)$
Window and Transport to Unit Scale
Orthonormal Ridgelet Analysis

Figure 1: Overview of Organization of the Curvelet Transform.

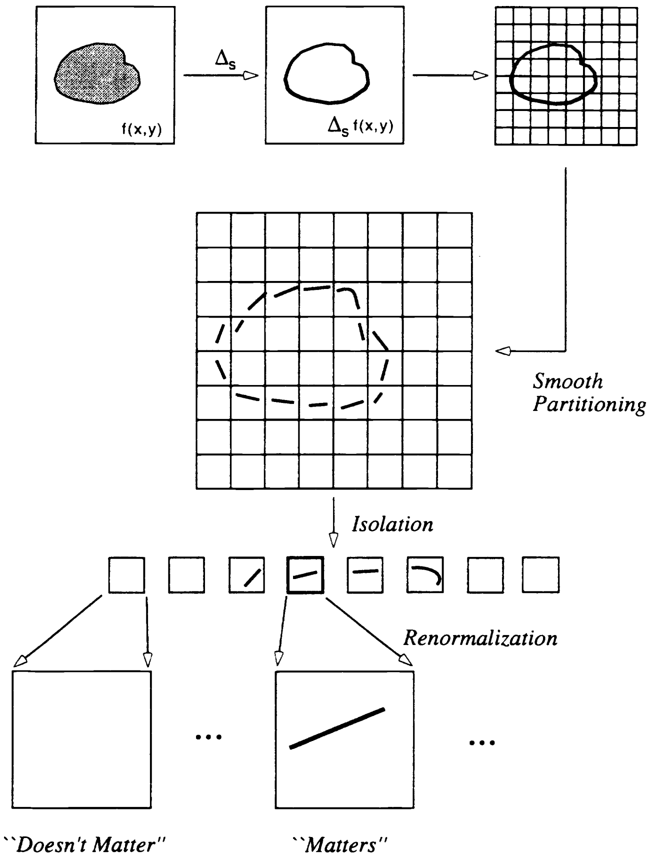

Figure 2: Spatial Decomposition of a Single Subband

The flow of this procedure is illustrated in Figure 1.

For an understanding of why the procedure might be organized as it is, consider Figure 2 .

Suppose that we have an object f which exhibits an edge. Upon subband filtering, each resulting fine-scale subband output $\Delta_s f$ will contain a map of the edge in f, thickened out to a width 2^{-2s} according to the scale of the subband filter operator. This gives the subband the appearance of a collection of smooth ridges. When we smoothly partition each subband into 'squares', we see either an 'empty square' – if the square does not intersect the edge – or a ridge fragment. Moreover, the ridge fragments are nearly straight at fine scales, because the edge is nearly straight at fine scales. Such nearly straight ridge fragments are precisely the desired input for the ridgelet transform.

2.2 Synthesis

There is also procedural definition of the reconstruction algorithm.

- *Ridgelet Synthesis.* Each 'square' is reconstructed from the orthonormal ridgelet system.
$$g_Q = \sum_\lambda \alpha_{(\lambda,Q)} \rho_\lambda$$
- *Renormalization.* Each 'square' resulting in the previous stage is renormalized to its own proper square
$$h_Q = (T_Q) g_Q, \qquad Q \in \mathcal{Q}_s.$$
- *Smooth Integration.* We reverse the windowing dissection to each of the windows reconstructed in the previous stage of the algorithm.
$$\Delta_s f = \sum_{Q \in \mathcal{Q}_s} w_Q \cdot h_Q.$$
- *Subband Recomposition.* We undo the bank of subband filters, using the reproducing formula:
$$f = P_0(P_0 f) + \sum_{s>0} \Delta_s(\Delta_s f).$$

2.3 Crucial Subtleties

2.3.1 Exact Reconstruction and Tight Frames

In the above procedures, the windowing w_Q and the filtering Δ_s underlying this procedure were specially constructed to insure that all these steps result in perfect reconstruction, and, in addition, a Parseval relation. Hence the window function is a nonnegative smooth function w, providing a partition of energy:

$$\sum_{k_1,k_2} w^2(x_1 - k_1, x_2 - k_2) \equiv 1, \qquad \forall (x_1, x_2).$$

Thus, if we form $h_Q = w_Q \cdot h$ for all $Q \in \mathcal{Q}_s$, then we have the exact reconstruction property

$$\sum_{Q \in \mathcal{Q}_s} w_Q \cdot h_Q = \sum_{Q \in \mathcal{Q}_s} w_Q^2 h = h,$$

while at the same time we have the energy-conservation property

$$\sum_{Q \in \mathcal{Q}_s} \|h_Q\|_2^2 = \sum_{Q \in \mathcal{Q}_s} \int w_Q^2 h^2 = \int \sum_{Q \in \mathcal{Q}_s} w_Q^2 h^2 = \int h^2 = \|h\|_2^2.$$

The subband filtering is based on the same idea, only in the frequency domain. We build a sequence of filters Φ_0 and $\Psi_{2s} = 2^{4s}\Psi(2^{2s}\cdot)$, $s = 0, 1, 2, \dots$ with the following properties: Φ_0 is a lowpass filter concentrated near frequencies $|\xi| \leq 1$; Ψ_{2s} is bandpass, concentrated near $|\xi| \in [2^{2s}, 2^{2s+2}]$; and we have the partition of energy property

$$|\hat{\Phi}_0(\xi)|^2 + \sum_{s \geq 0} |\hat{\Psi}(2^{-2s}\xi)|^2 = 1, \quad \forall \xi.$$

Then $P_0 f = \Phi_0 \star f$ and $\Delta_s f = \Psi_{2s} \star f$.

2.3.2 The Scaling Law

The precise definition of the bandpass filtering was given immediately above, and it contains one very noteworthy feature. The s-th subband is based on keeping frequencies in the corona $|\xi| \in [2^{2s}, 2^{2s+2}]$. The $2s$ in the exponent is different than what one might expect based on the s subscript Δ_s.

The point of this distinction is that *subband s contains ridges of width $\approx 2^{-2s}$ but is being sectioned into squares of side $\approx 2^{-s}$*. Therefore the resulting 'squares' which interact with edges will be ridge fragments of *width* $\approx 2^{-2s}$ and *length* $\approx 2^{-s}$; i.e. the aspect ratio obeys *width* $\approx$ *length*^{2}.

This highly anisotropic shape of the support is absolutely crucial to the performance of the transform; in particular the traditional isotropic scaling relation *length* $\approx$ *width* would take away all benefit of the subsequent step.

2.3.3 The Ridgelet Transform

Figure 3 helps illustrate a key point about the quantitative performance of the procedure. When one isolates a ridge fragment from subband s with aspect ratio 2^{-s} by 2^{-2s}, and renormalizes, one obtains an object which, in the frequency domain, has support localized to the frequency band $|\xi| \approx 2^s$, and lives in a region of width ≈ 1.

The final stage of the analysis procedure uses orthonormal ridgelets [11] to analyze such a fragment. These have an expression in the frequency domain as follows:

$$\hat{\rho}_\lambda(\xi) = |\xi|^{-\frac{1}{2}}(\hat{\psi}_{j,k}(|\xi|) w_{i,\ell}^\epsilon(\theta) + \hat{\psi}_{j,k}(-|\xi|) w_{i,\ell}^\epsilon(\theta + \pi))/2 .$$

Here the $\psi_{j,k}$ are Meyer wavelets for $\mathbf{R}$, $w_{i,\ell}^\epsilon$ are periodic wavelets for $[-\pi, \pi)$, and indices run as follows: $j, k \in \mathbf{Z}$, $\ell = 0, \dots, 2^{i-1} - 1$; $i \geq 1$, and, if $\epsilon = 0$, $i = \max(1, j)$, while if $\epsilon = 1$, $i \geq \max(1, j)$. We let $\lambda = (j, k, i, \ell, \epsilon)$ and let Λ be the set of such λ. Figuratively speaking, the ridgelet with $\lambda = (j, k, j, \ell, 0)$ has support in $|\xi| \approx 2^j$ and with a width

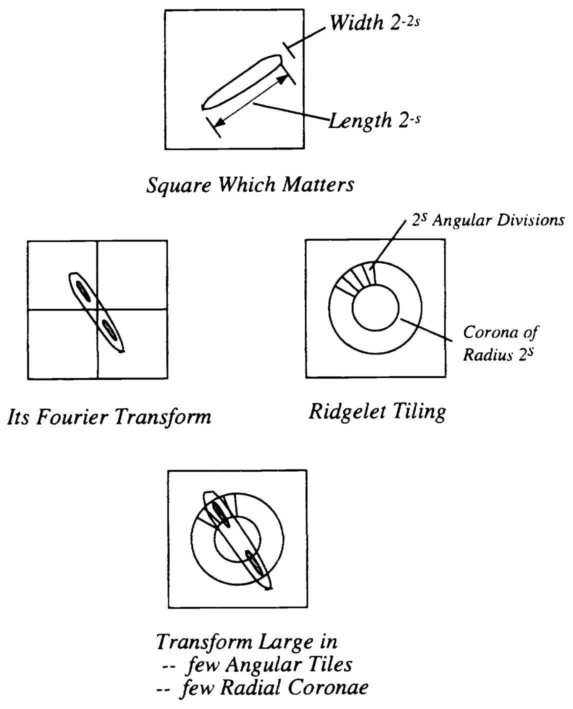

Figure 3: Illustration of Ridgelet Analysis of a Ridge Fragment.

of about 2^{-j}, localized near $2\pi\ell/2^j$. Hence, when taking ridgelet coefffficients, we are essentially overlaying on the image of the Fourier transform of f a sampling grid that is based on a collection of rectangular cells defined in polar coordinates.

Effectively, the idea behind the ridgelet transform is that when one encounters an object with a Fourier transform looking like such a ridge fragment, a very few ridgelet coefficients will be needed to represent it.

We note that it would not be very helpful to use classical transforms for such ridge fragments. The Fourier transform uses sinusoids, which correspond to points in the frequency domain. A ridge fragment's Fourier transform is again a ridge of dimensions 2^s by 1. Hence order 2^s coefficients are needed to represent a single ridge fragment using sinusoids. The Wavelet transform has elements which correspond to annular rings in the frequency domain, multiplied by sinusoids; their angular support is very large, effectively constant, indpendent of scale. The ridge fragment is supported in a band of angular resolution $O(2^{-s})$. Hence it also takes order 2^s coefficients to represent a single ridge fragment. Only the ridgelet basis has the required angular localization to mimic the ridge fragment signatures. For rigorous analysis, see the references, for example, [5].

3 Implementation Strategy

We now describe a strategy for realizing a digital implementation of the curvelet transform.

3.1 Specific Assumptions

Our strategy is based on a series of assumptions.

- *Image Size 256 * 256.* We have preferred to deal with images of size n by n, where $n = 2^8 = 4^4$. The choice of n a power of 4 is very important to our viewpoint. The strategy we are following would adapt most easily to the construction of transforms for $n = 1024$ and $n = 4096$. Accomodation to other sizes is possible.

- *Subband Definitions.* We have partitioned the frequency domain into only 3 subbands, indexed by $s = 1, 2, 3$. It is helpful to keep in mind that on a $256 * 256$ image, the usual discrete wavelet transform would offer 8 subbands, at levels $j = 0, \ldots, 7$. The Curvelet Subband $s = 1$ corresponds to wavelet subbands $j = 0, 1, 2, 3$ in a way we will describe later. Curvelet Subband $s = 2$ corresponds to wavelet subbands $j = 4, 5$, and Subband $s = 3$ corresponds to wavelet subbands $j = 6, 7$.

  The general rule of succession we are trying to implement in this way is

  $$\text{Curvelet subband } s \leftrightarrow \text{Wavelet Subbands } j \in \{2s, 2s + 1\}.$$

  Hence, in an implementation with $n = 4096$, we would have 5 subbands, with $s = 4$ corresponding to $j = 8, 9$ and $s = 5$ corresponding to $j = 10, 11$.

  These choices are responsible for implementing the anisotropic scaling principle underlying the curvelet transform, which is in some sense only an asymptotic principle. Thus, we have no objection in practice to adjustments to this set of subband definitions.

- *Subband Filtering.* To actually implement our decomposition into subbands, we use the wavelet transform. We first decompose the object into its 8 wavelet subbands, then to form Curvelet Subband s, we perform partial reconstruction from those wavelets at levels $j \in \{2s, 2s + 1\}$. Call the resulting 256*256 array $\tilde{D}_s$.

  It may turn out to be important that this is not actually an implementation of true subband filtering, but only an approximation; see remarks in Section 6.1.

- *Spatial Windowing.* To subband array $\tilde{D}_s$, we apply a localization into squares according to windows $\tilde{w}_Q$ which are of width about twice the width of the associated dyadic square.

  To give precise details, it is convenient to keep in mind two scaling conventions. In the *continuum convention* we are dealing with $Q = Q(s, k_1, k_2)$ defined as earlier, and spatial positions refer to points (x_1, x_2) in the square $[0, 1]^2$. In the *pixel convention* we have an array extending from 1 through 256 in each coordinate. The link is of course that spatial position (x_1, x_2) corresponds to pixel position (i_1, i_2) via $x_\ell = (i_\ell - 1)/256$. For example, the continuum dyadic square $Q(4, 5, 7) = [5/2^4, 6/2^4) \times [7/2^4, 8/2^4]$ corresponds most naturally to the pixel dyadic square $\tilde{Q}(4, 5, 7, 256) = [5 \cdot 2^4 + 1, 6 \cdot 2^4] \times [7 \cdot 2^4 + 1, 8 \cdot 2^4]$. The general formula puts $I_{s,k} = k \cdot n/2^s$ and

  $$\tilde{Q}(s, k_1, k_2, n) = [I_{s,k_1} + 1, I_{s,k_1+1}] \times [I_{s,k_2} + 1, I_{s,k_2+1}].$$

The window $\tilde{w}_Q$ is supported at samples (i_1, i_2) in the discrete square

$$[I_{s,k_1} - n/2^{s+1}, I_{s,k_1+1} + n/2^{s+1}] \times [I_{s,k_2} - n/2^{s+1}, I_{s,k_2+1} + n/2^{s+1}]$$

which is the "doubling" of the corresponding $\tilde{Q}$ about its center. The windows are designed to partition energy:

$$\sum_{k_1,k_2} \tilde{w}_Q(i_1, i_2)^2 = 1 \qquad \forall\, 1 \le i_1, i_2 \le 256.$$

In our implementation, we partitioned subbands into squares $\tilde{g}_Q$ in the expected way, with a twist. For $s = 2, 3$, we partition the subband D_{s-1} by dyadic squares in $\tilde{\mathcal{Q}}_s$:

$$\tilde{g}_Q(i_1 - I_{s,k_1}, i_2 - I_{s,k_2}) = \tilde{w}_Q(i_1, i_2) \cdot (\tilde{D}_{s-1})(i_1, i_2);$$

each such pixel array will be supported in pixels $[-n/2^{s+1}, 3n/2^{s+1}]^2$ (i.e. the support extends to negative indices in our convention).

Thus, in the pixel convention, the role of 'renormalization to the unit square' is replaced by 'clipping' to a square of side 2^{s+1} by 2^{s+1} pixels, and *in the implementation we have tried*, the linkage between $\Delta_s f$ and $\mathcal{Q}_s$ is substituted for a linkage between $\tilde{D}_{s-1}$ and $\tilde{\mathcal{Q}}_s$.

We emphasize this last detail. In the case where $n = 256$ which principally occupies us, this means that subband $s = 2$ is subdivided into an *eight-by-eight* array of squares, each supported in an array of 64×64, while subband $s = 3$ is subdivided into a *sixteen-by-sixteen* array of squares, each supported in an array of 32×32.

The seemingly more straightforward correspondence, between subband $\tilde{D}_s$ and $\tilde{\mathcal{Q}}_s$ (note the agreement of subscripts), which mimics notationally the definition in the continuum case, would result in a factor of two coarser subdivision of the image than the one we have adopted. In that correspondence subband $s = 2$ would be divided into a four-by-four array of squares. However, it seemed to us that for the experiments we had in mind, this degree of spatial partitioning was too coarse; the choice we adopted, $\tilde{D}_{s-1}$ and $\tilde{\mathcal{Q}}_s$, uses squares twice as fine.

It is in the choice of this correspondence that we impose the *width* $\approx$ *length*^{2} scaling. Since it is only an asymptotic notion, the factor of two does not make any substantial difference to the degree of faithfulness of the principle, while making better practical sense to us.

We suspect that in a larger image, we would use a continuation of this $s - 1 \leftrightarrow s$ calibration, so that subband $s = 4$ would involve a 32 by 32 array of 'squares' and $s = 5$ would involve a 64 by 64 array of 'squares'.

- *Digital Ridgelet Transform.* The innermost step of our algorithm is to apply the digital ridgelet transform to each "square" $\tilde{g}$. We use the DRT described in [12]; see also [1]. That transform has the following characteristics. Given an array of size $n \times n$, where n is dyadic, it returns an array of size $n \times 2n$ containing ridgelet coefficients. The transform is modeled on the orthonormal ridgelet transform described above, but the digital realization is not sufficiently faithful to be orthonormal. Instead,

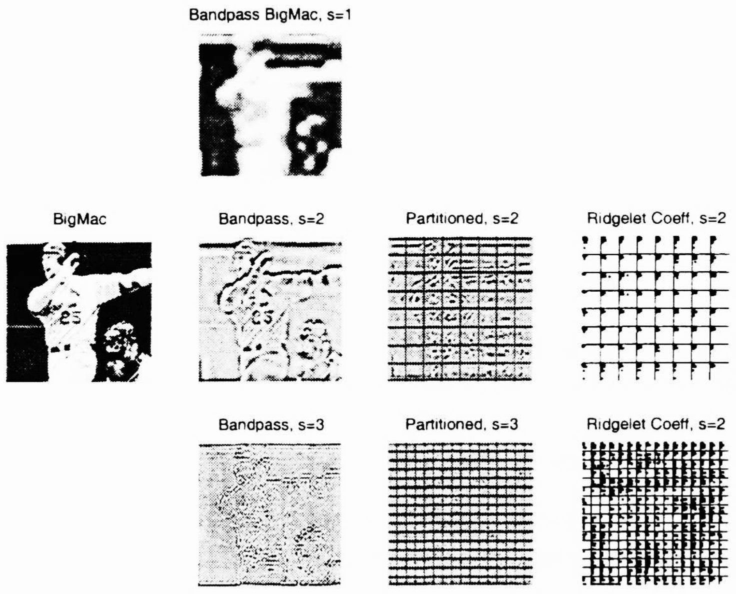

Figure 4: BigMac Image, and stages of Curvelet Analysis.

it provides a linear transform from $\ell^2(n^2)$ into $\ell^2(2n^2)$ which nearly preserves the norm of the transform. In fact, the ratio between the extreme singular values of the transform is about 2, so that it obeys a Parseval relation to within about a factor of two.

3.2 Example

We now give an example with the image BigMac, stolen by Donoho from a wire services web site in August 1999.

Figure 4 displays the stages of DCvT, the input data are at the extreme left, and processing moves us from left to right. The format is loosely patterned like Figure 1, so that the different subimages are located in positions corresponding to their appearance in the flow diagram Figure 1.

At the extreme left of this Figure is the BigMac image. The next column displays the three bandpass-filtered subbands. The next column displays the result of smoothly partitioning into 'squares'. The final column displays the ridgelet coefficients of each such square.

Figure 5 displays the stages of inverse DCvT, the input curvelet coefficients are at the extreme left, and processing moves us from left to right.

At the extreme left of this Figure are 1% of the curvelet coefficients of the BigMac image. All other coefficients have been set to zero. The next column displays the reconstruction of individual 'squares' from these local expansions. The next column displays the superposition of squares to yield a partial reconstruction of the corresponding subband. The final column displays the superposition of subband reconstructions into an overall reconstruction.

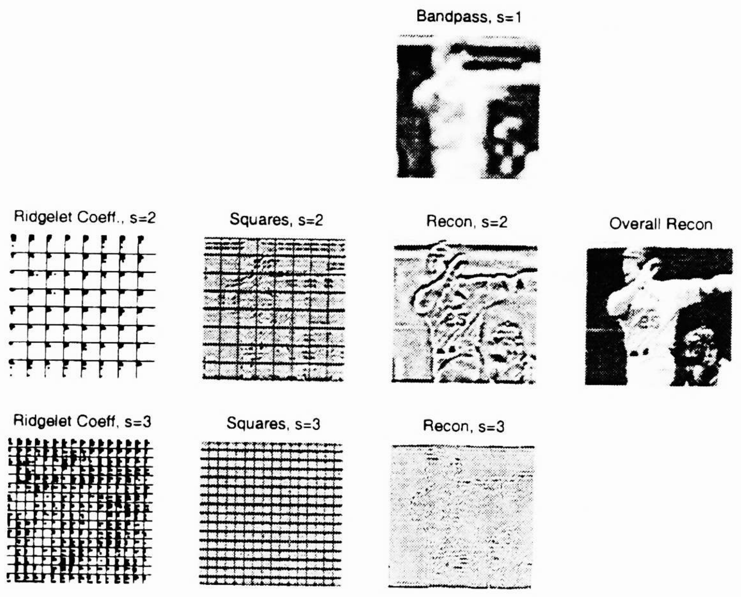

Figure 5: Curvelet Coefficients, and stages of Curvelet Synthesis.

3.3 Data Structures

We now summarize the data structures created by the above processing chain.

The transform of a 256-by-256 image has three arrays, containing the data associated with subbands $s = 1$, 2, and 3 respectively.

The output for Curvelet Subband $s = 1$ consists of Wavelet Subbands $j = 0, 1, 2, 3$; hence there are $(2^4)^2$ or 256 coefficients stored for this subband.

The output for Subband $s = 2$ is an array of extent $512 \times 512 \times 2$. Viewing each 512 by 512 slice as an image, one has embedded in this image an 8*8 array of 64*64 "squares".

The final output, for Subband $s = 3$ is an array of extent $512 \times 512 \times 2$. Viewing each 512 by 512 slice as an image, one has embedded in this image an 8*8 array of 64*64 "squares".

3.4 Expansiveness

The transform we have just described is highly expansive. A 256 by 256 image consisting of 64K pixels generates about 1M coefficients. This gives an expansion by a factor of about 16.

This can easily be understood by reviewing the chain of processing stages that go into the curvelet transform. We are concatenating several processing stages, and some of these are expansive by factors of 2 or even four, so the result is a total data volume sixteen times larger than the original.

The key expansive steps are:

- *Smooth windowing.* When performed on Subband 3, this takes an array of size 256*256 and breaks it into a 16*16 array of overlapping squares of size 32*32. The data in each resulting 32*32 square is 'associated to' a corresponding 16*16 'input square'. In other words, each 'output square' has four times as many numbers as the corresponding 'input square'. So this step is expansive by a factor 4.

- *Ridgelet Transform.* The digital ridgelet transform we are using takes an n by n array and returns an n by $2n$ array. So this stage is expansive by a factor 2.

There appears to be substantial opportunities for additional decimation. The bandpass filtering stage removes frequencies outside a certain band, but we do not exploit this fact when we later take the ridgelet transform. In particular, the ridgelet transform ought to be essentially vanishing at ridge scales j far from s. By a careful matching of the bandpass operator and the ridgelet transform, we ought to be able to arrange that all coefficients outside a certain scale (or interval of scales) be omitted both from calculation and from storage.

It appears that this lost opportunity is reponsible for an additional factor 2 expansionism.

4 Curvelet256 Toolbox

We have built a Toolkit of Matlab routines that carries out the above strategy, and allows us to conduct experiments. The idea of this toolkit was to enable us to try out a set of standard curvelet transform domain manipulates on a suite of image data.

The toolkit is designed so that one can assimilate a 256*256 image into the given framework, run some standard scripts, and generate some `.mat` files containing curvelet transform coefficients along with others containing various partial reconstructions. Among the processing tasks one can try are 'movie making', in which a sequence of frames illustrates the progressive reconstruction of an image by using successively more curvelet terms.

An online version of this article is available at

`http://www-stat.stanford.edu/~donoho/Reports/2000/curvsoft.pdf`

in Acrobat format (.ps and .ps.Z files are also available). In that version of the article several pages describe in detail the steps required to download, install, and use the software.

5 Image Processing Experiments

We have now applied the above tools, in a variety of settings, to the following datasets:

Barbara	Classic for less Sexist Image Processors
BigMac	Mark McGwire Hits One Out
Brain	MRI of someone's Brain
fprint	Fingerprint
InTheCar	Roy Lichtenstein Cartoon (1960's)
Lenna	Classic for Sexist Image Processors
Picasso	'Three Women' Engraving (1922-23)
Siesta	Millet Painting (1850's)
Tube	A Sinusoid
Yukon	Satellite Imagery of Mountains

Some of our results can be found on the web, at URL

`http://www-stat.stanford.edu/~donoho/curvelet.site`

This web site makes available several MPEG movies that can be viewed with any standard web browser. Some of these movies illustrate that curvelet reconstructions, based on a very few coefficients, can provide a very good idea of the geometry of the underlying object. Particularly good movies to view include

- The movie of the fingerprint image `fprint` shows that reconstruction from just a few hundred coefficients give a very clear idea of the fingerprint geometry. The movie is located at URL

  `http://www-stat.stanford.edu/~donoho/curvelet.site/fprint.mov`

- The movie of the cartoon image `InTheCar` shows again that a sense of the edge features can be obtained very rapidly, with just a few coefficients. The movie is located at URL

  `http://www-stat.stanford.edu/~donoho/curvelet.site/BigMac.mov`

As an example, we include here a set of four frames from the fingerprint movie. The first frame gives the $s = 1$ approximation using 64 (father) wavelets at the coarsest level; one has no inkling that the underlying image is a fingerprint. The second frame shows that adding just 256 curvelets immediately recaptures the geometry of the fingerprint. The third frame shows more geometric information being blended in, and the final frame shows that most of the work at later stages is filling in texture on the ridges in the image.

We next display an engraving by Picasso,'Three Women', 1922-23. This engraving, from the permanent collection of the Metropolitan Museum of Art, was made during a period in which Picasso was experimenting with the idea of 'Drawings in one continuous line', an idea that gained currency through the work of André Breton and other postwar revolutionaries. The image is nearly a simple curvilinear sketch, almost executed in a single continuous line, although the background color is not uniformly flat. We display the

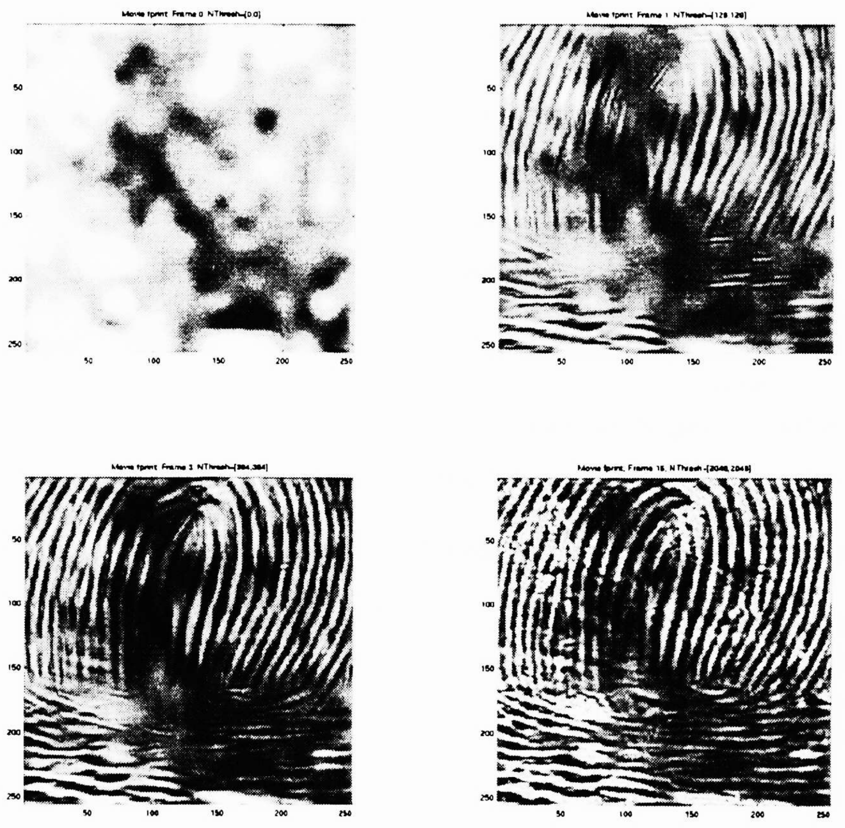

Figure 6: Four frames from the Fingerprint Movie. Panel (a) Approximation by 64 Wavelet coefficients; (b) Adding in 256 Curvelet Coefficients; (c) Adding another 768 Curvelet Coefficients; (d) Adding another 3072 Curvelet Coefficients

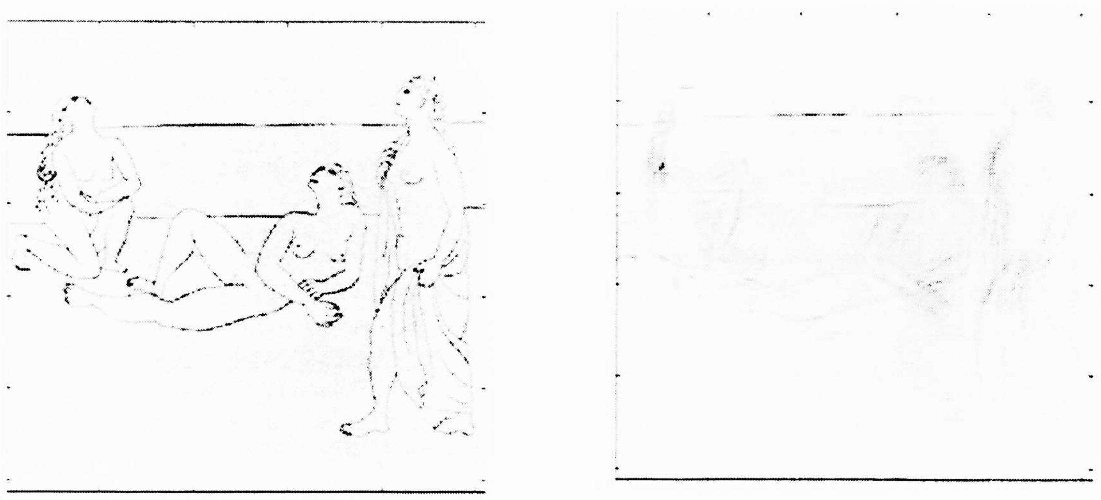

Figure 7: Picasso, 'Three Women'. Panel (a) Scanned Original, converted to square format; (b) Approximation by 256 Curvelet Coefficients.

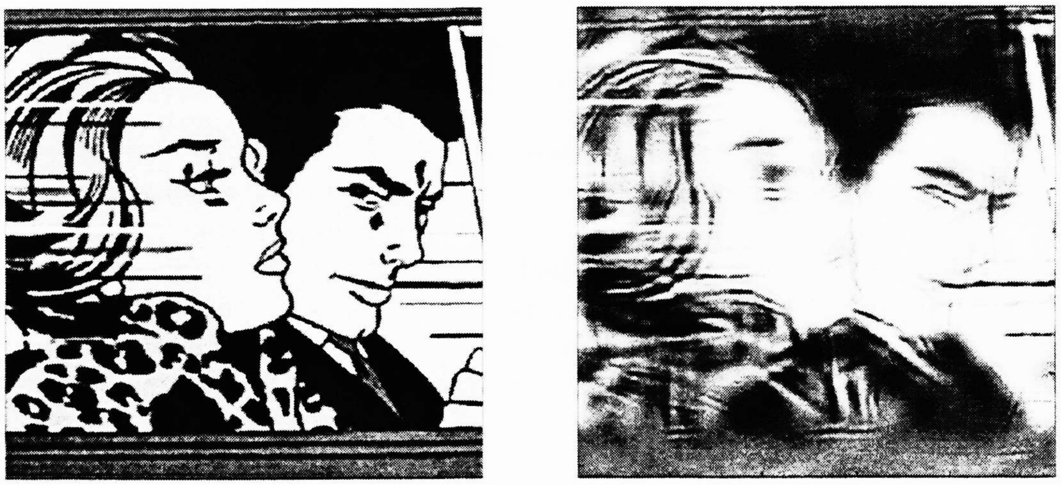

Figure 8: Lichtenstein, 'In The Car'. Panel (a) Scanned Original, converted to square format; (b) Approximation by 64 wavelets and 256 Curvelet Coefficients.

results of approximation from 256 curvelets at $s = 2, 3$. The curvelets begin to capture the geometry rather quickly.

Finally, we display a painting by Roy Lichtenstein, 'In The Car', from his series of 'Cartoon' paintings of the 1960's, key paintings from the Pop Art movement. We also display an approximation by 256 curvelets with $s = 2, 3$, and 64 wavelets at $s = 1$.

6 Directions for Future Work

We see several avenues for further exploration.

6.1 Directions for Improved Implementation

We are hardly satisfied with the performance of our existing DCvT scheme. On the one hand, working with the raw transform is clumsy because of the factor 16 expansivity. In fact, each subband has more coefficients than the original image has samples. On the other hand, while the transform makes rapid progress towards reconstructing the object as the first few coefficients 'paint in' the geometric structure like so many well-chosen brushstrokes, after a few thousand coefficients have been added, the progress towards the ultimate reconstruction slows down substantially.

It seems to us that the "expansivity problem" needs to be addressed by extensive practical and theoretical work. On the other hand, the issue of "slowing progress" seems addressible by smallish modifications of the basic software. We discuss two such modifications.

6.1.1 Better Ridgelets

The ridgelet transform we have used suffers from a positional aliasing problem. If the desired shape of a 'logical ridgelet' is of an elongated sausage or needle, then a display of the frame elements of our underlying digital ridgelet transform has the appearance of pairs of antipodal 'logical ridgelets'. This 'twinning' effect is undesirable, and suggests that in any expansion where, logically, one ridgelet would do the trick, at least two will have to be used, in order to cancel one of the two partners in a pair. If this effect could be repaired, the achievable compression ratio might double.

The explanation of this effect seems subtle, we believe it has to do with the reliance of the DRT algorithm on fast fourier transforms and on the underlying toroidal periodicity of the FFT.

In our opinion, this effect may be remedied by modifying the transform slightly, so that instead of providing an n by 2n transform, it provides a 2n by 2n transform. However, this step would increase in expansivity by a factor of 2, and so doing this in a naive way would increase the expansivity of the overall DCvT from a factor 16 to a factor of 32, which is clearly in the wrong direction.

6.1.2 Better Bandpass Filters

The bandpass filtering we have used is in actuality not traditional bandpass filtering at all, but instead what we call 'wavelet bandpass filtering'. In order to get an image localized to frequencies near the band $[2^{2s}, 2^{2s+2})$ we literally expand the object in nearly-symmetric Daubechies wavelets and discard terms except at $j = 2s$ or $j = 2s + 1$.

This approach is convenient from a software development standpoint, since the required wavelet tools are available in Wavelab. However, this pseudo-bandpass filtering injects directional artifacts into the output. If there is an edge at θ radians, one often sees in the pseudo-bandpass output a ghost edge at $\theta + \pi/2$ radians. This 'twinning' is again undesirable since it again suggests that in an expansion where, logically, one ridgelet would do the trick, at least two will have to be used, in order to cancel one of the two partners in a pair.

In our opinion, a stricter adherence to the spirit of the continuous transform would help here. By performing a true frequency-domain bandpass filtering, the directional aliasing can be avoided.

6.1.3 Better Decimation

The output of subband filtering is in principle bandlimited. It would seem that a representation of the coefficients with substantially smaller expansivity might be developed based on this. In effect, rather than applying the full ridgelet transform to each square, one would extract only coefficients at subbands where nonzero results are logically possible. This might substantially reduce storage requirements for all subbands except the very finest scales.

6.2 Directions for Fundamental Research

6.2.1 A Search for New Refinement Schemes

A novel and interesting aspect of the curvelet scheme is the fact that each generation of refinement leads to a doubling of the spatial resolution as well as a doubling of the angular resolution. This aspect – where the number of resolvable feature directions increases with scale – is very different from wavelet and associated approaches.

In the algorithm presented here, we have explored a frequency-side approach to defining curvelets. An interesting open question: is there a spatial-domain approach, starting from the above structural feature? That is, is there a spatial domain scheme for refinement which, at each generation doubles the spatial resolution as well as the angular resolution?

6.2.2 Understanding Sampling

A fundamental problem facing the project is the fact that *pixel sampling is very problematic* for phenomena *where angular sensitivity is important.* If we may be permitted poetic language, it seems to us that 'most' of the 'effort' in reconstructing an object during progressive curvelet reconstruction is 'caused' by the necessity to make the reconstruction and image match at a pixel level along edges. Because of pixelization, the underlying property of real imagery – that while images are abruptly discontinuous across edges, but smooth along edges – is completely disrupted by pixelization: digital images are oscillatory along the edge even when the true underlying image is smooth along the edge. This phenomenon restricts the effectiveness of curvelet representation of digital data; using once again poetic language, the digital curvelets are 'trying to represent' the underlying continuum image rather than the pixelized one.

Clearly this issue calls for much deeper understanding.

References

[1] Averbuch, A., Coifman, R.R., Donoho, D.L., Israeli, M., and Waldén, J. (1999) RectoPolar FFT and its Applications. Manuscript.

[2] Candès, E. (1999) Harmonic Analysis of Neural Networks, Appl. Comput. Harmon. Anal. **6** (1999), 197–218.

[3] Candès, E. (1998) *Ridgelets: Theory and Applications.* Ph.D. Thesis, Department of Statistics, Stanford University.

[4] Candès, E. (1999) Monoscale ridgelets for the representation of images with edges, Technical Report, Statistics, Stanford.

[5] Candès, E. (1999) On the Representation of Mutilated Sobolev Functions. Technical Report, Statistics, Stanford.

[6] Candès, E. and Donoho, D. (1999) Ridgelets: The key to High-Dimensional Intermittency?. Phil. Trans. R. Soc. Lond. A. **357** (1999), 2495-2509

[7] Candès, E. and Donoho, D. (1999) Curvelets. Manuscript.

[8] Candès, E. and Donoho, D. (1999) Curvelets: a surprisingly effective nonadaptive representation for objects with edges. in *Curves and Surfaces IV* ed. P.-J. Laurent.

[9] Candès, E. and Donoho, D. (1999) Curvelets and Linear Inverse Problems. Manuscript.

[10] Candès, E. and Donoho, D. (1999) Curvelets and Curvilinear Integrals. Manuscript.

[11] Donoho, D. (1998) Orthonormal Ridgelets and Linear Singularities. To appear, *SIAM J. Math. Anal.*

[12] Donoho, D. (1998) Digital Ridgelet Transform via Digital Polar Coordinate Transform. Manuscript.

[13] M. Frazier, B. Jawerth, and G. Weiss (1991) *Littlewood-Paley Theory and the study of function spaces.* NSF-CBMS Regional Conf. Ser in Mathematics, **79**. American Math. Soc.: Providence, RI.

[14] P.G. Lemarié and Y. Meyer. (1986) Ondelettes et bases hilbertiennes. *Rev. Mat. Iberoamericana* **2**, 1-18.

[15] Meyer, Y. (1993) *Wavelets: Algorithms and Applications.* Philadelphia: SIAM.

[16] Meyer, Y. (1993) Review of *An Introduction to Wavelets* by Charles Chui. *Bull. Amer. Math. Soc. (N.S.)* **28** 350-360.