3D Discrete Curvelet Transform

Lexing Ying, Laurent Demanet and Emmanuel Candès

Applied and Computational Mathematics, MC 217-50, Caltech, Pasadena, CA

ABSTRACT

In this paper, we present the first 3D discrete curvelet transform. This transform is an extension to the 2D transform described in Candès et al..[1] The resulting curvelet frame preserves the important properties, such as parabolic scaling, tightness and sparse representation for singularities of codimension one. We describe three different implementations: in-core, out-of-core and MPI-based parallel implementations. Numerical results verify the desired properties of the 3D curvelets and demonstrate the efficiency of our implementations.

Keywords: Curvelet transform; Parabolic scaling; High-dimensional singularities; Partition of unity; Fast Fourier transform; Out-of-core algorithm; Parallel algorithm; Wrapping.

1. INTRODUCTION

Curvelets are two dimensional waveforms that provide a new architecture for multiscale analysis. In space, a curvelet at scale j is an oriented "needle" whose effective support is a 2^{-j} by $2^{-j/2}$ rectangle and thus obeys the parabolic scaling relation $width \approx length^2$. In frequency, a curvelet at scale j is a "wedge" whose frequency support is again inside a rectangle, but of size 2^j by $2^{j/2}$. Unlike wavelets, curvelets are localized not only in position (the spatial domain) and scale (the frequency domain), but also in orientation. This localization provides the curvelet frame with surprising properties: it is an optimally sparse representation for singularities supported on C^2 curves in two dimensions[2]; it forms an optimally sparse basis for pseudodifferential operators (PDOs) and Fourier integral operators (FIOs)[3,4] and it has become a promising tool for various image processing problems.[5]

Starck et al.[5] gave the first discrete curvelet transform specialized for image processing applications. However, in the past few years, curvelets have been redesigned to make them easy to use and understand.[2] Recently, Candès et al.[1] presented two 2D discrete curvelet transforms for the second-generation curvelets. These new transforms are numerically tight frames, and the resulting curvelets are faithful analogs of their continuous counterparts.

In many scientific and engineering disciplines, such as video processing, seismic imaging and medical imaging, the data is inherently three dimensional. With the increase of computational power, direct processing of three dimensional data becomes feasible. Since properties such as scale-position-direction localization and parabolic scaling can naturally be extended to 3D, a 3D curvelet transform will open new opportunities to analyze and study large data sets in these fields.

In this paper, we present the first 3D discrete curvelet transform. This transform is an extension to the 2D transform presented in Candès et al..[1] This new discrete curvelet frame preserves the important properties, such as parabolic scaling, tightness and sparse representation for surface-like singularities of codimension one.

The paper is organized as follows. Section 2 reviews the 2D curvelet transform. Section 3 briefly outlines the 2D discrete transform. In Section 4, we describe the architecture of the 3D discrete curvelet transform. Section 5 presents three different implementations of the 3D transform and some numerical results.

Contact info: [lexing,demanet,emmanuel]@acm.caltech.edu.

2. 2D CURVELET TRANSFORM

In this section, we briefly introduce the curvelet transform in 2D. We work throughout in $\mathbf{R}^2$, with spatial variable x, with frequency value ω, and with r and θ being the polar coordinates in the frequency domain. We start with a pair of windows $W(r)$ and $V(t)$, which we will call the *radial window* and the *angular window*, respectively. They are both smooth, nonnegative and real-valued, with W taking positive real arguments and supported on $r \in [1/2, 2]$ and V taking real arguments and supported on $t \in (-2\pi, 2\pi]$. These windows will always obey the conditions:

$$\sum_{j=-\infty}^{\infty} W^2(2^{-j}r) = 1, \qquad r > 0,$$

$$\sum_{\ell=-\infty}^{\infty} V^2(t - 2\pi\ell) = 1, \qquad t \in \mathbf{R}.$$

As in the wavelet theory, we introduce the lowpass window W_{j_0} which satisfies the following condition

$$W_{j_0}(r)^2 + \sum_{j>j_0} W(2^{-j}r)^2 = 1.$$

For each $j > j_0$, $W(2^{-j}r)$ smoothly extracts the frequency content inside the dyadic region $(2^{j-1}, 2^{j+1})$. Curvelets are organized by the triple index (j, ℓ, k), with j standing for scale, ℓ for orientation, and $k = (k_1, k_2)$ for spatial location.

Coarsest scale $j = j_0$. The coarsest scale curvelets are isotropic, and the only index for ℓ is zero. We define the frequency window $U_{j_0,0}$ by

$$U_{j_0,0}(\omega) = W_{j_0}(\omega).$$

Suppose $U_{j_0,0}$ is supported on a rectangle of size L_{1,j_0} by L_{2,j_0} (by our choice of W_{j_0}, $L_{1,j_0} = L_{2,j_0}$). The coarsest curvelets are defined by means of their Fourier transform

$$\hat{\varphi}_{j_0,0,k}(\omega) = U_{j_0,0}(\omega) \cdot \exp[-2\pi\mathrm{i}(k_1\omega_1/L_{1,j_0} + k_2\omega_2/L_{2,j_0})]/\sqrt{L_{1,j_0} \cdot L_{2,j_0}}.$$

Fine scale $j > j_0$. The frequency content radially extracted by $W(2^{-j}r)$ is further partitioned into $2^{j/2}$ angular windows. For each $\ell : 0 \le \ell < 2^{j/2}$, we define a wedge-like frequency window $U_{j,\ell}(\omega)$ by

$$U_{j,\ell}(\omega) = W(2^{-j}r) \cdot V(2^{j/2}(\theta - \theta_\ell))$$

where $\theta_\ell = 2\pi\ell \cdot 2^{-j/2}$ (see Figure 1(a)). Suppose $U_{j,0}$ is supported a rectangle of size $L_{1,j}$ by $L_{2,j}$. Clearly $L_{1,j} = O(2^j)$ and $L_{2,j} = O(2^{j/2})$. The curvelets at scale j and orientation $\ell = 0$ are defined through their Fourier transforms

$$\hat{\varphi}_{j,0,k}(\omega) = U_{j,0}(\omega) \cdot \exp[-2\pi\mathrm{i}(k_1\omega_1/L_{1,j} + k_2\omega_2/L_{2,j})]/\sqrt{L_{1,j} \cdot L_{2,j}}.$$

Directly from this definition, we know, for any $k = (k_1, k_2)$ with $k_1, k_2 \in \mathbf{Z}$, $\varphi_{j,0,k}(x) = \varphi_{j,0,0}(x - (k_1/L_{1,j}, k_2/L_{2,j}))$. For general ℓ, the curvelets of orientation ℓ are defined by

$$\varphi_{j,\ell,k}(x) = \varphi_{j,0,k}(R_{-\theta_\ell} \cdot x)$$

where R_θ is the rotation matrix by θ radians. Obviously, the Fourier transform $\hat{\varphi}_{j,\ell,k}$ of a curvelet $\varphi_{j,\ell,k}$ is localized on the support of the frequency window $U_{j,\ell}$.

The curvelet coefficients of a function $f \in L^2(\mathbf{R}^2)$ are simply the inner products between f and the curvelets $\varphi_{j,\ell,k}$

$$c(j, \ell, k) := \langle \varphi_{j,\ell,k}, f \rangle = \int_{\mathbf{R}^2} \overline{\varphi_{j,\ell,k}(x)} f(x)\, dx.$$

The coarsest scale curvelets are non-directional. However, it is the behavior of the fine-scale directional elements that are of interest. Figure 1 summarizes the key components of this construction. We now summarize a few properties of the curvelet transform:

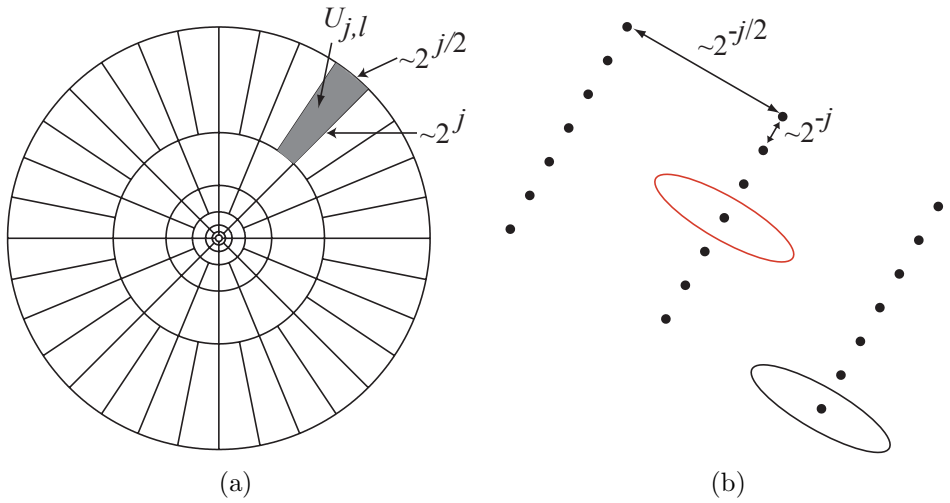


Figure 1. 2D curvelet tiling of space and frequency. (a) The induced tiling of the frequency domain. In Fourier space, curvelets are supported near a parabolic wedge, $U_{j,\ell}$ is one of these wedges. (b) The associated Cartesian grid in space. The spacing of the grid again obeys the parabolic scaling.

1. **Tight-frame**. Much like in an orthonormal basis, we can expand an arbitrary function $f(x_1, x_2) \in L^2(\mathbf{R}^2)$ as a series of curvelets: we have a reconstruction formula

$$f = \sum_{j,\ell,k} \langle \varphi_{j,\ell,k}, f \rangle \varphi_{j,\ell,k},$$

with equality holding in an L^2 sense; and a Parseval relation

$$\sum_{j,\ell,k} |\langle f, \varphi_{j,\ell,k} \rangle|^2 = \|f\|^2_{L^2(\mathbf{R}^2)}, \quad \forall f \in L^2(\mathbf{R}^2).$$

2. **Parabolic scaling**. The frequency localization of $U_{j,\ell}$ implies the following spatial structure: $\varphi_{j,\ell,k}(x)$ is of rapid decay away from a 2^{-j} by $2^{-j/2}$ rectangle with major axis orthogonal to the direction θ_ℓ. In short, the effective length and width obey the scaling relation

$$length \approx 2^{-j/2}, \quad width \approx 2^{-j} \quad \Rightarrow \quad width \approx length^2.$$

3. **Oscillatory behavior**. As is apparent from its definition, $U_{j,0}$ is actually supported away from the vertical axis $\omega_1 = 0$ but near the horizontal $\omega_2 = 0$ axis. This implies that $\varphi_{j,0,k}(x)$ is oscillatory in the x_1-direction and lowpass in the x_2-direction. The situation for any other $\varphi_{j,\ell,k}$ is exactly the same up to a rotation. Hence, at scale j, a curvelet is a fine needle whose envelope is a ridge of effective length $2^{-j/2}$ and width 2^{-j}, and which displays an oscillatory behavior across its minor axis.

4. **Optimal basis for curve-like singularities**. As a result of the parabolic scaling property, the curvelet frame is the optimal sparse representation for those functions with singularities along C^2 curves but otherwise smooth.[4]

3. 2D DISCRETE TRANSFORM

In this section, we describe the 2D discrete curvelet transform on which our 3D transform is based. The discrete transform takes as input a Cartesian grid of the form $f(n_1, n_2)$, $0 \le n_1, n_2 < n$, and outputs a collection of coefficients $c^D(j, \ell, k)$ defined by

$$c^D(j, \ell, k) := \sum_{n_1, n_2} f(n_1, n_2) \, \overline{\varphi^D_{j,\ell,k}(n_1, n_2)}$$

where $\varphi^D_{j,\ell,k}$ are digital curvelet waveforms which preserve the listed properties of the continuous curvelet. The definition of these discrete curvelets $\varphi^D_{j,\ell,k}$ are parallel to the definition of their continuous counterparts.

The frequency grid of the input f is $\hat{f}(\omega)$ with $\omega = (\omega_1, \omega_2)$ and $-n/2 \le \omega_1, \omega_2 < n/2$. In the continuous definition, the window $U_{j,\ell}$ smoothly extracts frequencies near the dyadic corona $\{2^{j-1} \le r \le 2^{j+1}\}$ and near the angle $\{-\pi \cdot 2^{-j/2} \le \theta - \theta_\ell \le \pi \cdot 2^{-j/2}\}$. Coronae and rotations are not adapted to Cartesian data. Instead, it is convenient to replace these concepts by Cartesian equivalents: the "Cartesian coronae" based on concentric squares (instead of circles) and shears. The Cartesian analog to the family $(W_j)_{j \ge j_0}$, $W_j(\omega) = W(2^{-j}\omega)$, is a window of the form

$$\widetilde{W}_{j_0}(\omega) = \Phi_{j_0}(\omega) \quad \text{and} \quad \widetilde{W}_j(\omega) = \sqrt{\Phi^2_{j+1}(\omega) - \Phi^2_j(\omega)}, \quad j \ge j_0,$$

where Φ is defined as the product of low-pass one dimensional windows $\Phi_j(\omega_1, \omega_2) = \phi(2^{-j}\omega_1) \cdot \phi(2^{-j}\omega_2)$. The function ϕ is smooth, obeys $0 \le \phi \le 1$, is equal to 1 on $[-1, 1]$ and vanishes outside of $[-2, 2]$. It is immediate to check that

$$\widetilde{W}_{j_0}(\omega)^2 + \sum_{j \ge j_0} \widetilde{W}^2_j(\omega) = 1.$$

To angularly window each Cartesian corona, we introduce a smooth and localized function $\widetilde{V}$ which satisfies

$$\sum_{\ell=-\infty}^{\infty} \widetilde{V}^2(t - 2\ell) = 1, \qquad t \in \mathbf{R}.$$

Coarsest scale $j = j_0$. The frequency window $\widetilde{U}_{j_0,0}$ is defined by

$$\widetilde{U}_{j_0,0}(\omega) = \widetilde{W}_{j_0}(\omega).$$

Suppose $U_{j_0,0}$ is supported in a rectangle of integer size L_{1,j_0} by L_{2,j_0}. Discrete curvelets at the coarsest level are defined by their discrete Fourier transforms:

$$\hat{\varphi}^D_{j_0,0,k}(\omega) = \widetilde{U}_{j_0,0}(\omega) \cdot \exp[-2\pi \mathrm{i}(k_1\omega_1/L_{1,j_0} + k_2\omega_2/L_{2,j_0})]/\sqrt{L_{1,j_0} \cdot L_{2,j_0}}$$

for $0 \le k_1 < L_{1,j_0}$ and $0 \le k_2 < L_{2,j_0}$.

Fine scale $j_0 < j < j_e$. Each Cartesian coronae has four quadrants: East, North, West and South. Each quadrant is separated into $2^{j/2}$ wedges with the same areas (see Figure 2(a)). Take the East quadrant ($-\pi/4 \le \theta \le \pi/4$) as an example. Suppose we order the wedges in a counterclockwise way. The center slope for the ℓth wedge $\alpha_\ell = -1 + 2 \cdot (\ell + 1/2) \cdot 2^{-j/2}$. We define the smooth angular window for the ℓth direction as

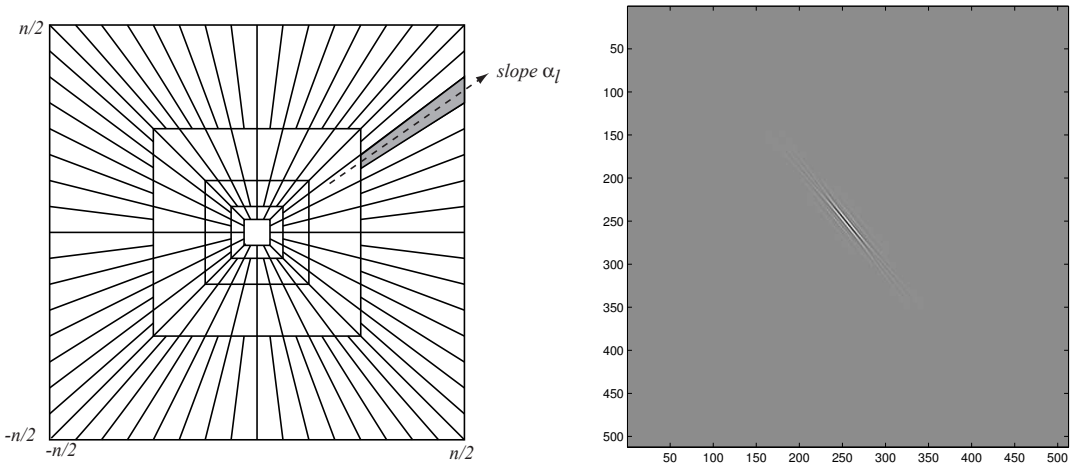
$$\widetilde{V}_{j,\ell}(\omega) = \widetilde{V}(2^{j/2} \cdot \frac{\omega_2 - \alpha_\ell \cdot \omega_1}{\omega_1}).$$

For the other quadrants, we would have a similar definition by exchanging the roles of ω_1 and ω_2. Based on the definition of $\widetilde{V}_{j,\ell}$, each ω is covered by exactly two of the angular windows $\widetilde{V}_{j,\ell}$ from scale j. The sum of the squares of $\widetilde{V}_{j,\ell}$ for a fixed j is equal to one except for ω near the diagonals ($|\omega_1| = |\omega_2|$). To enforce this partition of unity property, which is essential for the tightness of the discrete transform, the following step is sufficient. Suppose two smooth angular windows $\widetilde{V}_{j,\ell}$ and $\widetilde{V}_{j,\ell'}$ overlap, but the sum of their square is not equal to one on the overlapping region of their supports. We redefine them on the overlapping region by:

$$\left(\widetilde{V}_{j,\ell}(\omega), \widetilde{V}_{j,\ell'}(\omega)\right) \leftarrow \frac{1}{\sqrt{\widetilde{V}^2_{j,\ell}(\omega) + \widetilde{V}^2_{j,\ell'}(\omega)}} \left(\widetilde{V}_{j,\ell}(\omega), \widetilde{V}_{j,\ell'}(\omega)\right)$$

The frequency window $\widetilde{U}_{j,\ell}$ is then defined by

$$\widetilde{U}_{j,\ell}(\omega) = \widetilde{W}_j(\omega) \cdot \widetilde{V}_{j,\ell}(\omega).$$



(a) (b)

Figure 2. 2D discrete curvelet transform. (a) Discrete frequency tiling. $\widetilde{U}_{j,\ell}$ has center slope α_ℓ. It smoothly localizes the frequency near the shaded wedge. (b) One curvelet at scale j and orientation ℓ in spatial domain. Notice that the major axes of the curvelet in the frequency and space domains are orthogonal to each other.

It is clear that $\widetilde{U}_{j,\ell}$ isolates frequencies near the wedge $\{(\omega_1,\omega_2) : 2^{j-1} \le \omega_1 \le 2^{j+1}, -2^{-j/2} \le \omega_2/\omega_1 - \alpha_\ell \le 2^{-j/2}\}$.

With the localized frequency window $\widetilde{U}_{j,\ell}$ available, the final step is to choose a spatial grid to translate the curvelet at scale j and orientation ℓ. In the continuous transform, the grid we use has its two axes aligned with the major and minor axes of the frequency window. For the discrete transform, two approaches are possible: (1) a slanted grid mostly aligned with the axes of the frequency window which leads to the USFFT-based curvelet transform (for details, see Candès at al[1]); (2) a grid aligned with the input Cartesian grid which leads to the wrapping-based curvelet transform. Here we follow the wrapping-based approach.

Fix the scale j and angle ℓ. Suppose $L_{1,j,\ell}$ and $L_{2,j,\ell}$ are a pair of positive integers which satisfy the following conditions: (1) one cannot find two ω and ω' such that $\widetilde{U}_{j,\ell}(\omega) \ge 0$, $\widetilde{U}_{j,\ell}(\omega') \ge 0$, and $\omega_1 - \omega_1'$ and $\omega_2 - \omega_2'$ are multiples of $L_{1,j,\ell}$ and $L_{2,j,\ell}$ respectively; and (2) $L_{1,j,\ell} \cdot L_{2,j,\ell}$ is minimal.

The discrete curvelet with index k at scale j and angle ℓ is defined by means of its Fourier transform:

$$\hat{\varphi}^D_{j,\ell,k}(\omega) = \widetilde{U}_{j,\ell}(\omega) \cdot \exp[-2\pi \mathrm{i}(k_1\omega_1/L_{1,j,\ell} + k_2\omega_2/L_{2,j,\ell})]/\sqrt{L_{1,j,\ell} \cdot L_{2,j,\ell}}.$$

for $0 \le k_1 < L_{1,j,\ell}$ and $0 \le k_2 < L_{2,j,\ell}$. Geometrically, the computation of the coefficients $\varphi^D_{j,\ell,k}$ for fixed j and ℓ is equivalent to wrapping the windowed frequency data $\widetilde{U}_{j,\ell}(\omega)\hat{f}(\omega)$ around a $L_{1,j,\ell}$ by $L_{2,j,\ell}$ rectangle centered at the origin, and then applying the inverse FFT to the wrapped data. This justifies the word "wrapping". Our choice of $L_{1,j,\ell}$ and $L_{2,j,\ell}$ guarantees the data does not overlap with itself after the wrapping process.

Last scale $j = j_e = \log_2(n/2)$. This final scale extracts the highest frequency content. For the purpose of this paper, the basis functions used at this scale are like wavelets (for other choices, see Candès et al[1]). The frequency window is

$$\widetilde{U}_{j_e,0}(\omega) = \widetilde{W}_{j_e}(\omega).$$

The curvelets at this level are defined by

$$\hat{\varphi}^D_{j_e,0,k}(\omega) = \widetilde{U}_{j_e,0}(\omega) \cdot \exp[-2\pi \mathrm{i}(k_1\omega_1/L_{1,j_e} + k_2\omega_2/L_{2,j_e})]/\sqrt{L_{1,j_e} \cdot L_{2,j_e}},$$

with $L_{1,j_e} = L_{2,j_e} = n$ and $0 \le k_1, k_2 < n$.

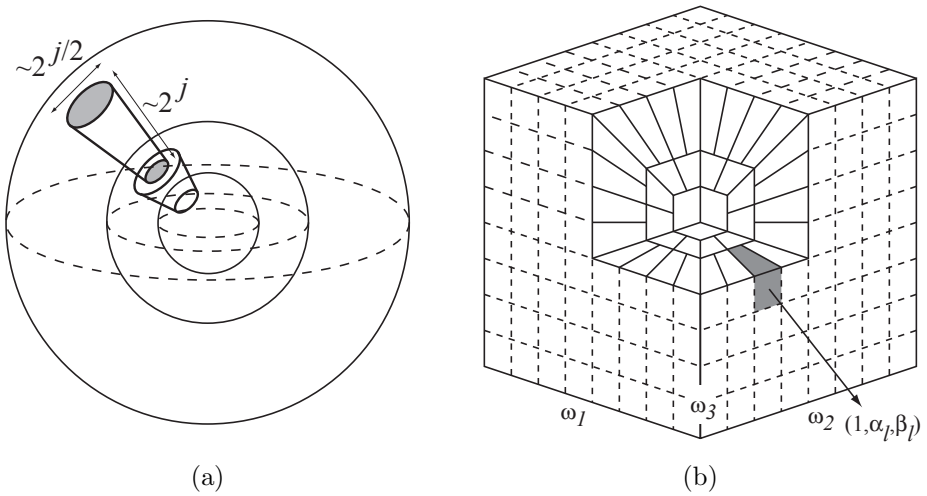


Figure 3. 3D frequency tilings. (a) Schematic plot for the frequency tiling of continuous 3D curvelets. (b) Discrete frequency tiling. ω_1, ω_2 and ω_3 are three axes of the frequency cube. Smooth frequency window $\widetilde{U}_{j,\ell}$ extracts the frequency content near the shaded wedge which has center slope $(1, \alpha_\ell, \beta_\ell)$.

This frame of discrete curvelets has all the required properties of the continuous curvelet transform in Section 2. Figure 2(b) shows one typical curvelet in the spatial domain. To summarize, the algorithm of the 2D discrete curvelet transform is as follows:

1. Apply the 2D FFT and obtain Fourier samples $\hat{f}(\omega_1, \omega_2)$, $-n/2 \le \omega_1, \omega_2 < n/2$.
2. For each scale j and angle ℓ, form the product $\widetilde{U}_{j,\ell}(\omega_1, \omega_2)\hat{f}(\omega_1, \omega_2)$.
3. Wrap this product around the origin and obtain $\mathcal{W}(\widetilde{U}_{j,\ell}\hat{f})(\omega_1, \omega_2)$, where the range for ω_1 and ω_2 is now $-L_{1,j,\ell}/2 \le \omega_1 < L_{1,j,\ell}/2$ and $-L_{2,j,\ell}/2 \le \omega_2 < L_{2,j,\ell}/2$. For $j = j_0$ and j_e, no wrapping is required.
4. Apply a $L_{1,j,\ell} \times L_{2,j,\ell}$ inverse 2D FFT to each $\mathcal{W}(\widetilde{U}_{j,\ell}\hat{f})$, hence collecting the discrete coefficients $c^D(j, \ell, k)$.

4. 3D DISCRETE CURVELET TRANSFORM

The 3D curvelet transform is expected to preserve the properties of the 2D transform. Most importantly, the frequency support of a 3D curvelet shall be localized near a wedge which follows the parabolic scaling property. One can prove that this implies that the 3D curvelet frame is a sparse basis for representing functions with surface-like singularities (which is of codimension one in 3D) but otherwise smooth. For the continuous transform, we window the frequency content as follows. The radial window smoothly extracts the frequency near the dyadic corona $\{2^{j-1} \le r \le 2^{j+1}\}$, this is the same as the radial windowing used in 2D. For each scale j, the unit sphere S^2 which represents all the directions in $\mathbf{R}^3$ is partitioned into $O(2^{j/2} \cdot 2^{j/2}) = O(2^j)$ smooth angular windows, each of which has a disk-like support with radius $O(2^{-j/2})$, and the squares of which form a partition of unity on S^2 (see Figure 3(a)).

Like the 2D discrete transform, the 3D discrete curvelet transform takes as input a 3D Cartesian grid of the form $f(n_1, n_2, n_3), 0 \le n_1, n_2, n_3 < n$, and outputs a collection of coefficients $c^D(j, l, k)$ defined by

$$c^D(j, \ell, k) := \sum_{n_1, n_2, n_3} f(n_1, n_2, n_3)\, \overline{\varphi^D_{j,\ell,k}(n_1, n_2, n_3)}$$

where $j, \ell \in \mathbf{Z}$ and $k = (k_1, k_2, k_3)$.

We use Cartesian coronae of the form

$$\widetilde{W}_{j_0}(\omega) = \Phi_{j_0}(\omega) \quad \text{and} \quad \widetilde{W}_j(\omega) = \sqrt{\Phi^2_{j+1}(\omega) - \Phi^2_j(\omega)}, \quad j \geq j_0,$$

with $\Phi_j(\omega_1, \omega_2, \omega_3) = \phi(2^{-j}\omega_1) \cdot \phi(2^{-j}\omega_2) \cdot \phi(2^{-j}\omega_3)$, where the function ϕ is the same as before.

Coarsest Scale $j = j_0$. The frequency window $\widetilde{U}_{j_0,0}$ is again defined by

$$\widetilde{U}_{j_0,0}(\omega) = \widetilde{W}_{j_0}(\omega).$$

Suppose $U_{j_0,0}$ is supported in a rectangular box of integer size $L_{1,j_0} \times L_{2,j_0} \times L_{3,j_0}$. The discrete curvelets at the coarsest level are defined by their Fourier transforms:

$$\hat{\varphi}^D_{j_0,0,k}(\omega) = \widetilde{U}_{j_0,0}(\omega) \cdot \exp[-2\pi \mathrm{i}(k_1\omega_1/L_{1,j_0} + k_2\omega_2/L_{2,j_0} + k_3\omega_3/L_{3,j_0})]/\sqrt{L_{1,j_0} \cdot L_{2,j_0} \cdot L_{3,j_0}}$$

for $0 \leq k_1 < L_{1,j_0}$, $0 \leq k_2 < L_{2,j_0}$ and $0 \leq k_3 < L_{3,j_0}$.

Fine Scale $j_0 < j < j_e$. Now every Cartesian corona has six components, one for each face of the unit cube. Each component is regularly partitioned into $2^{j/2} \cdot 2^{j/2}$ wedges with same volume (see Figure 3(b)). Take the first component (corresponding to $w_1 > 0$) as an example. Suppose, for the ℓth wedge, $(1, \alpha_\ell, \beta_\ell)$ is the direction of the center line of the wedge. We define its smooth angular window as

$$\widetilde{V}_{j,\ell}(\omega) = \widetilde{V}(2^{j/2} \cdot \frac{\omega_2 - \alpha_\ell \cdot \omega_1}{\omega_1}) \cdot \widetilde{V}(2^{j/2} \cdot \frac{\omega_3 - \beta_\ell \cdot \omega_1}{\omega_1})$$

where $\widetilde{V}$ is defined as before. For the other five components, we would have a similar definition by exchanging the roles of ω_1 with ω_2 or ω_3.

Similar to the 2D case, the squares of $\widetilde{V}_{j,\ell}$ form a partition of unity except near ω satisfying $|\omega_i| = |\omega_j|$ for $i \neq j$. We enforce this property everywhere by using the modification applied to the 2D case. Suppose three smooth angular windows $\widetilde{V}_{j,\ell}$, $\widetilde{V}_{j,\ell'}$ and $\widetilde{V}_{j,\ell''}$ overlap, but the sum of their squares is not equal to one on the overlapping region of their supports. We redefine them on the overlapping region using:

$$\left(\widetilde{V}_{j,\ell}(\omega), \widetilde{V}_{j,\ell'}(\omega), \widetilde{V}_{j,\ell''}(\omega)\right) \leftarrow \frac{1}{\sqrt{\widetilde{V}^2_{j,\ell}(\omega) + \widetilde{V}^2_{j,\ell'}(\omega) + \widetilde{V}^2_{j,\ell''}(\omega)}} \left(\widetilde{V}_{j,\ell}(\omega), \widetilde{V}_{j,\ell'}(\omega), \widetilde{V}_{j,\ell''}(\omega)\right)$$

The frequency window $\widetilde{U}_{j,\ell}$ is then defined by

$$\widetilde{U}_{j,\ell}(\omega) = \widetilde{W}_j(\omega) \cdot \widetilde{V}_{j,\ell}(\omega).$$

It is clear that $\widetilde{U}_{j,\ell}$ isolates frequencies near the wedge $\{(\omega_1, \omega_2, \omega_3) : 2^{j-1} \leq \omega_1 \leq 2^{j+1}, -2^{-j/2} \leq \omega_2/\omega_1 - \alpha_\ell \leq 2^{-j/2}, -2^{-j/2} \leq \omega_3/\omega_1 - \beta_l \leq 2^{-j/2}\}$ (see Figure 3(b)).

The final step is to translate the discrete curvelet. Fix the scale j and angle ℓ. Suppose $L_{1,j,\ell}$, $L_{2,j,\ell}$ and $L_{3,j,\ell}$ are three positive integers such that: (1) one cannot find two ω and ω' such that $\widetilde{U}_{j,\ell}(\omega) \geq 0$, $\widetilde{U}_{j,\ell}(\omega') \geq 0$, and $\omega_1 - \omega'_1$, $\omega_2 - \omega'_2$ and $\omega_3 - \omega'_3$ are multiples of $L_{1,j,\ell}$, $L_{2,j,\ell}$ and $L_{3,j,\ell}$ respectively; and (2) $L_{1,j,\ell} \cdot L_{2,j,\ell} \cdot L_{3,j,\ell}$ is minimal. The discrete curvelets with index k at scale j and angle ℓ are defined by means of their Fourier transforms:

$$\hat{\varphi}^D_{j,\ell,k}(\omega) = \widetilde{U}_{j,\ell}(\omega) \cdot \exp[-2\pi \mathrm{i}(k_1\omega_1/L_{1,j,\ell} + k_2\omega_2/L_{2,j,\ell} + k_3\omega_3/L_{3,j,\ell})]/\sqrt{L_{1,j,\ell} \cdot L_{2,j,\ell} \cdot L_{3,j,\ell}}$$

for $0 \leq k_1 < L_{1,j,\ell}$, $0 \leq k_2 < L_{2,j,\ell}$ and $0 \leq k_3 < L_{3,j,\ell}$.

Last scale $j = j_e = \log_2(n/2)$. This final scale extracts the highest frequency content of the 3D frequency cube. Similar to the 2D discrete transform, the basis functions here are again wavelet-like. The frequency window is

$$\widetilde{U}_{j_e,0}(\omega) = \widetilde{W}_{j_e}(\omega).$$

The curvelets at this level are defined by

$$\hat{\varphi}^D_{j_e,0,k}(\omega) = \widetilde{U}_{j_e,0}(\omega) \cdot \exp[-2\pi \mathrm{i}(k_1\omega_1/L_{1,j_e} + k_2\omega_2/L_{2,j_e} + k_3\omega_2/L_{3,j_e})]/\sqrt{L_{1,j_e} \cdot L_{2,j_e} \cdot L_{3,j_e}},$$

with $L_{1,j_e} = L_{2,j_e} = L_{3,j_e} = n$ and $0 \le k_1, k_2, k_3 < n$.

To summarize, the algorithm 3D discrete curvelet transform is as follows:

1. Apply the 3D FFT and obtain Fourier samples $\hat{f}(\omega_1, \omega_2, \omega_3)$, $-n/2 \le \omega_1, \omega_2, \omega_3 < n/2$.
2. For each scale j and angle ℓ, form the product $\widetilde{U}_{j,\ell}(\omega_1, \omega_2, \omega_3)\hat{f}(\omega_1, \omega_2, \omega_3)$.
3. Wrap this product around the origin and obtain $\mathcal{W}(\widetilde{U}_{j,\ell}\hat{f})(\omega_1, \omega_2, \omega_3)$, where the range for ω_1, ω_2 and ω_3 is now $-L_{1,j,\ell}/2 \le \omega_1 < L_{1,j,\ell}/2$, $-L_{2,j,\ell}/2 \le \omega_2 < L_{2,j,\ell}/2$ and $-L_{3,j,\ell}/2 \le \omega_3 < L_{3,j,\ell}/2$. No wrapping is necessary at scales j_0 and j_e.
4. Apply a $L_{1,j,\ell} \times L_{2,j,\ell} \times L_{3,j,\ell}$ 3D inverse FFT to each $\mathcal{W}(\widetilde{U}_{j,\ell}\hat{f})$, hence collecting the discrete coefficients $c^D(j, \ell, k)$.

Figure 4 shows how the 3D curvelets look like in the spatial domain. There are plate-like objects which follow parabolic scaling and oscillates along the shortest axis. So far, we have discussed the 3D discrete curvelet transform based on the wrapping strategy. The 2D USFFT-based transform can also be extended to 3D with minor modifications.[1]

5. IMPLEMENTATIONS

A problem we encounter in the 3D transform is the drastic increase of the size of the data. To store a complex-valued Cartesian grid of size 512^3 with double-precision, we need 2GB of memory. Moreover, the curvelet transform is redundant in general, and the 3D transform described in Section 4 has a redundancy factor of about 5. It is impossible for most computers to store data of this size in the memory. This imposes extra complications on the implementation of a 3D curvelet transform for large data from real scientific and engineering applications. In this section, we propose three different implementations differing by how the data is managed.

In-core implementation. This is a direct implementation of the transform described in the previous section. Both the input data and the curvelet coefficients are stored in the memory. This implementation is extremely efficient: each 3D curvelet transform takes the same time as a few 3D FFT transform. The disadvantage of this implementation is obviously the limitation on the size of the input grid. A common desktop computer with 1GB memory can only handle Cartesian grids of size below 200^3. Table 1 shows the performance of our algorithm with several input grids on a desktop computer with a 2.6GHz CPU. By denoting the forward and inverse transforms by C_F and C_I respectively, we measure the error by $\|f - C_I(C_F(f))\|_{l^2}/\|f\|_{l^2}$ for an input grid f. Since our transform is designed to be numerically tight, the error is close to the machine accuracy.

Grid size	Memory Used	Time	Error
$64 \times 64 \times 64$	40MB	1.4 sec	1.3055e-15
$128 \times 128 \times 128$	320MB	10.8 sec	1.4731e-15
$180 \times 180 \times 180$	900MB	34.8 sec	1.2213e-15

Table 1. Results of the in-core implementation.

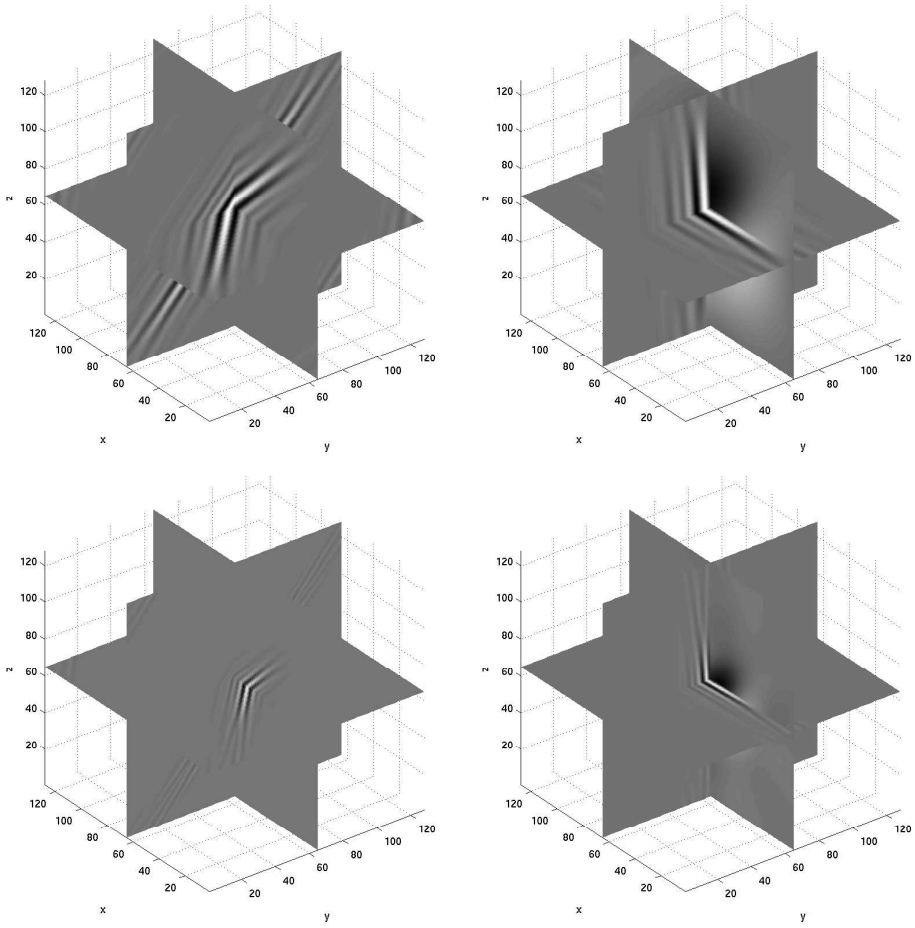


Figure 4. 3D discrete curvelets at two scales. In each plot, we show the behavior of the curvelet at three cross-sections. Top: two coarse scale 3D curvelets. Bottom: two fine scale 3D curvelets.

Out-of-core implementation. In this implementation, only the input Cartesian grid and a small portion of the curvelet coefficients are stored in memory, while most of the curvelet coefficients are stored on the disk. Whenever a curvelet coefficient is required, we check whether it is in the memory. If not, we load it from the disk. We also keep track of how many curvelet coefficients reside in the memory. Whenever the number is above a certain threshold, we release the memory used by some of the curvelet coefficients and store them onto the disk. Such coefficients are selected based on the Least-Recently-Used (LRU) algorithm, which is commonly used for resource management by all kinds of computer systems. This implementation has the advantage that we can reuse the existing in-core FFT algorithms on the input Cartesian data. It is able to handle a grid of size 500^3 on a desktop computer. Table 2 presents the performance of our out-of-core algorithm on two medium-size data sets.

Grid size	Memory Used	Time
$256 \times 256 \times 256$	768MB	250 sec
$512 \times 512 \times 512$	6GB	2100 sec

Table 2. Results of the out-of-core implementation.

MPI-based Parallel implementation. In this implementation, both the input Cartesian grid and the curvelet coefficients are stored distributed on multiple processors. The Cartesian grid is partitioned into small cubes of size $b \times b \times b$. Suppose n is a multiple of b and define $p = n/b$. Throughout the MPI-based transform, the coefficients in one cube are treated atomically: they are loaded, stored, discarded and communicated altogether. At any moment, every processor has a subset of these cubes residing in its memory and the union of these subset over all processors contains all the cubes.

Curvelet coefficients are naturally organized into wedges, each one corresponding to a frequency window $\widetilde{U}_{j,\ell}$. The partition of these wedges among the processors maximizes the coherence: the wedges belonging to one processor are geometrically close to each other in the frequency domain.

The parallel transform has the following steps:

1. Partition the $b \times b \times b$ cubes of the input grid into slices along the z direction. Each processor has a few adjacent slices in its memory.
2. Apply the $n \times n \times n$ 3D FFT on the input data. This is done in three stages, each one containing the 1D FFTs in one Cartesian axis. After the first two stages in the x and y directions, a communication step migrates the cubes among the processors to make the data ready for the third stage in the z direction.
3. Migrate the cubes again so that each process has in its memory the cubes overlapping with its wedges.
4. For each processor, iterate through its wedges. For each wedge (j, ℓ), form the product $\widetilde{U}_{j,\ell}\hat{f}$, wrap it to obtain $\mathcal{W}(\widetilde{U}_{j,\ell}\hat{f})$, and perform the 3D inverse FFT on $\mathcal{W}(\widetilde{U}_{j,\ell}\hat{f})$ to collect the curvelet coefficients $c^D(j, \ell, k)$. These inverse 3D FFTs are done locally.
5. For the last level, window the frequency data with $\widetilde{U}_{j_e,0}$, and perform the $n \times n \times n$ inverse 3D FFT to get the curvelet coefficients at the last level. This is similar to the second step.

Table 3 presents the result of our parallel implementation on several data sets. These experiments are performed on the IBM cluster "DataStar" at the San Diego Supercomputing Center. Each processor of this cluster has CPU speed 1.5GHz.

Grid size	Number of Processors	Time
$256 \times 256 \times 256$	4	47 sec
$512 \times 512 \times 512$	32	50 sec
$1k \times 1k \times 1k$	128	94 sec

Table 3. Results of the parallel implementation.

6. CONCLUSION

In this paper, we present the first 3D discrete curvelet transform. This transform is an extension to the 2D transform presented in Candès et al..[1] The resulting curvelet frame preserves the important properties, such as parabolic scaling, tightness and being a sparse basis for surface-like singularities of codimension one. We have presented three different implementations to address the memory issue caused by large data sets. All three implementations are highly efficient. We expect our 3D discrete transform to be widely applicable in various scientific and engineering fields.

ACKNOWLEDGMENTS

E. C. is partially supported by a National Science Foundation grant DMS 01-40698 (FRG) and by a Department of Energy grant DE-FG03-02ER25529. L. Y. is supported by a Department of Energy grant DE-FG03-02ER25529.

REFERENCES

1. E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying, "Fast discrete curvelet transforms," tech. rep., Applied and Computational Mathematics, California Institute of Technology, 2005.
2. E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Comm. Pure Appl. Math.* **57**(2), pp. 219–266, 2004.
3. E. Candès and L. Demanet, "Curvelets and Fourier integral operators," *C. R. Math. Acad. Sci. Paris* **336**(5), pp. 395–398, 2003.
4. E. J. Candès and L. Demanet, "The curvelet representation of wave propagators is optimally sparse," to appear in Communications on Pure and Applied Mathematics.
5. J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.* **11**(6), pp. 670–684, 2002.