Uniform Discrete Curvelet Transform

Truong T. Nguyen and Hervé Chauris

***Abstract*—An implementation of the discrete curvelet transform is proposed in this work. The transform is based on and has the same order of complexity as the Fast Fourier Transform (FFT). The discrete curvelet functions are defined by a parameterized family of smooth windowed functions that satisfies two conditions: i) 2π periodic; ii) their squares form a partition of unity. The transform is named the uniform discrete curvelet transform (UDCT) because the centers of the curvelet functions at each resolution are positioned on a uniform lattice. The forward and inverse transform form a tight and self-dual frame, in the sense that they are the exact transpose of each other. Generalization to M dimensional version of the UDCT is also presented. The novel discrete transform has several advantages over existing transforms, such as lower redundancy ratio, hierarchical data structure and ease of implementation.**

***Index Terms*—Contourlet, curvelet, directional decomposition, directional filter bank, multidimensional filter bank, multiresolution representation, wavelet.**

I. Introduction

During the last two decades, there have been a lot of research activities on new mathematical and computational tools for multiresolution data representation. The wavelet transform, arising from a particular question in geophysics, decomposes a function on the real line to a sum of local wave-like functions at multiple scales. It is shown that the continuous wavelet functions and their discrete counterpart implemented by regular filter banks are optimal representations of one dimensional piece-wise smooth signals [1]. However, the direct extension of a wavelet to two dimensions by the tensor product of two one-dimensional wavelets is no longer optimal for representing a signal that has features along smooth curves [2]. There have been many 2-D transforms with directional basis that can better represent this type of signal. Without being exhaustive, we list several examples of directional, discrete and nonadaptive transforms such as the steerable pyramid [3], the complex wavelet [4], [5], discrete curvelet [6], shearlet transforms [7], and contourlet transform [8].

Recently, Candès and Donoho constructed the curvelet transform [9], and proved that it is an essentially optimal representation of two-variable functions, which are smooth except at discontinuities along $\boldsymbol{C}^2$ (twice differentiable) curve. The nonlinear approximation of a function f, $f_M^{(c)}$, reconstructed by M largest curvelet coefficients has an asymptotic decay rate of $\left\|f - f_M^{(c)}\right\|^2 \le CM^{-2}(\log_2 M)^3$. This decay rate of the approximation error is a significant theoretical improvement compared to those by wavelets or Fourier coefficients, which are $O(M^{-1})$ and $O(M^{-1/2})$, respectively [1]. The 2-D curvelet transform decomposes a signal into a sum of basis functions that represent local waves with strong directionality. The effective support of these curvelet functions are highly anisotropic at fine scale, following a parabolic scaling rule: length^2 ~ width.

It has been proved that curvelet representation for wave propagation is optimally sparse [10]. It is argued that the representation of seismic data by the curvelet transform is also optimally sparse and a series of applications can be found in the area of seismic imaging [11]–[13]. The curvelet transform has also found applications in other areas of image processing [14]–[16].

In the last few years, several discrete curvelet and curvelet-like transforms have been proposed. They can be divided into discrete transforms based on the fast Fourier transform (FFT) [6], [17], or based on filter bank (FB) implementations [8], [18], [19].

The major contribution of this paper is a new discrete curvelet transform that uses the ideas of both FFT-based discrete curvelet transform and filter-bank based contourlet transform. The motivation of the new transform is from the difficulties we encountered while trying to use curvelet transform in real world applications. The new uniform discrete curvelet transform (UDCT) is implemented by the FFT algorithm, but it is designed as a multiresolution filter bank. By this way, we are able to take advantage of the two methods. For example, the new curvelet basis functions have excellent frequency responses, which is not possible with the contourlet transform. The UDCT has significantly lower redundancy compared to other FFT-based discrete curvelet transforms [6], [17], especially in high-dimension version of the transform. This makes the UDCT very practical in industrial applications.

Paper Outline: In the next section the notations used in this paper are introduced. Since the UDCT is interpreted as a multiresolution FB, this section also includes a short review of some formulae for multidimensional multirate systems. The curvelet and contourlet transforms are reviewed in Section III. The continuous curvelet transform is explained in term of sampling and reconstruction of band-limited signal. We discuss the typical examples of the two available approaches in implementing a discrete transform with curvelet-like frequency division, the fast discrete curvelet transform (FDCT) [6] and the contourlet transform [8]. After an analysis of the pros and cons of the two discrete transforms, we construct a discrete curvelet transform that combines the advantages of both methods in the rest of the

Manuscript received May 27, 2009; accepted February 21, 2010. Date of publication April 05, 2010; date of current version June 16, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Soontorn Oraintara. This work was supported in part by Shell E&P.

T. T. Nguyen is with the R&D Department, Fugro Seismic Imaging, Swanley, U.K. (e-mail: truong.nguyen@fugro-fsi.com).

H. Chauris is with the Centre de Géosciences, Mines ParisTech, UMR Sisyphe 7619, Fontainebleau, 77300 France (e-mail: herve.chauris@mines-paristech.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2010.2047666

paper. In Section IV, a parameterized family of smooth windowed functions is defined. These functions will later be used as curvelet functions in the frequency domain. They are all 2π periodic and their squares form a partition of unity, which means that the sum of squares of all functions is equal to 1 on the 2-D plane. The set of windows is then used in Section VI to define a multiresolution multidirectional FB. The discrete transform implemented by the FB is formally defined in Section VI. The proposed transform, namely the uniform discrete curvelet transform, is generalized to M-D in Section VII. The complexity of the transform and various implementation aspects are discussed in Section VIII. We present several numerical experiments using the UDCT in Section IX and conclude the paper in Section X. A Matlab package to implement the transform is available for download at http://sites.google.com/site/nttruong/.

II. Preliminary

Notation: In this paper, common multirate signal processing notations as in [21] are used. However, we also try to keep the same notation as in other works about discrete curvelet and contourlet transforms [6], [8], [22]. Since curvelet decomposition is a two dimensional transform, the functions in this paper are typically two-variable. We use bold letters and numbers to signify vectors, and bold capital letters to signify matrices. For example, $\boldsymbol{I} = \mathrm{diag}\{1,1\}$ is the 2×2 identity matrix. Bold letter variables signify functions of two or more variables. We denote by t the spatial variable in $\mathbb{R}$ and $\boldsymbol{t} = (t_1, t_2)$ the spatial variable in $\mathbb{R}^2$. In the Fourier domain, $\boldsymbol{\xi} = (\xi_1, \xi_2)$ is the 2-D frequency variable. $\boldsymbol{n} = (n_1, n_2)$ are coordinates of 2-D discrete functions (or signal). Similarly, $\boldsymbol{\omega} = (\omega_1, \omega_2)$ is discrete frequency variable. By an abuse of notation, r and θ are used as the polar coordinate of the $\boldsymbol{\xi}$ or $\boldsymbol{\omega}$ frequency planes. Throughout the paper, bold letter variables are employed frequently to compress mathematical expressions. For example, the following expressions are equivalent:

$$\sum_{n_1,n_2=-\infty}^{\infty} w_0(\omega_1 + 2n_1\pi, \omega_2 + 2n_2\pi) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} w_0(\boldsymbol{\omega} + 2\boldsymbol{n}\pi). \tag{1}$$

Small letters and capital letters denote a discrete filter in the spatial and frequency domains, for example $f(\boldsymbol{n})$ and $F(\boldsymbol{\omega})$. However, the hat symbol denotes the Fourier transform of continuous functions. For example the Fourier transform of $\varphi(\boldsymbol{t})$ is $\hat{\varphi}(\boldsymbol{\xi})$.

In the continuous domain, a curvelet basis function can be indexed by scale j, direction l and position $\boldsymbol{k}$. Therefore, the curvelet index set is $\mu = (j, l, \boldsymbol{k})$. However, the index for the new discrete curvelet transform is $\mu = (i, j, l, \boldsymbol{k})$ where i is the index for symmetric/antisymmetric property of the curvelet functions.

Grid and Lattice: A *grid* $\Lambda(\boldsymbol{M})$ is a set of points in a 2-D space generated by a nonsingular matrix $\boldsymbol{M}$, defined by

$$\Lambda(\boldsymbol{M}) = \{\boldsymbol{M}\boldsymbol{n}\}, \quad \boldsymbol{n} \in \mathbb{Z}^2 \tag{2}$$

In this paper, we use the word *lattice* to denote a grid in case all elements of $\boldsymbol{M}$ are integer.

Rotation and Shearing: A rotation of a 2-D function $f(\boldsymbol{t})$ by an angle θ is $f(\boldsymbol{R}_\theta \boldsymbol{t}) = f(\cos(\theta)t_1 + \sin(\theta)t_2, -\sin(\theta)t_1 + \cos(\theta)t_2)$. When the matrix $\boldsymbol{R}_\theta$ is replaced by an upper or lower 2×2 matrix with diagonal elements equal to 1, the new function is a shearing version of $f(\boldsymbol{t})$.

A. Discrete Signal Decimation and Upsampling, and Filter Bank

A discrete filter bank is a set of cascading discrete filters, decimation and upsampling blocks (see Fig. 10 as an example). If the elements of a FB are designed in such a way that the output $y(\boldsymbol{n})$ is exactly $x(\boldsymbol{n})$, then the FB corresponds to a discrete transform. Since we will use the family of windows $u_l^2(\omega_1, \omega_2)$ as discrete filters in the frequency domain of a FB, it is important to establish the equivalent operation in the frequency domain when a discrete signal passes through the decimation and upsampling blocks as in Fig. 2.

The relationship between a decimated signal $x_d(n)$ and the original signal $x(n)$ in the spatial domain is $x_d(mn) = x(n)$, where m is a natural number. The relationship between $X_d(\omega)$ and $X(\omega)$ is

$$X_d(\omega) = \frac{1}{m}\sum_{k=0}^{m-1} X\left(\frac{\omega - 2k\pi}{m}\right). \tag{3}$$

If a signal $x(n)$ is upsampled by m to produce $x_u(n) = x(n/m)$ if n/m is an integer, and $x_u(n) = 0$ otherwise, then in the frequency domain

$$X_u(\omega) = X(m\omega). \tag{4}$$

The two (3) and (4) are for 1-D signal. If a 2-D signal $x(\boldsymbol{n})$ is downsampled and then upsampled by 2 in two dimensions (or by $2\boldsymbol{I}$), the resulting signal $x_{du}(\boldsymbol{n}) = x(\boldsymbol{n})$ when $n \in \Lambda(2\boldsymbol{I})$, otherwise $x_{du}(\boldsymbol{n}) = 0$. Following (3) and (4), in the frequency domain

$$X_{du}(\boldsymbol{\omega}) = \frac{1}{4}\sum_{\boldsymbol{n}\in\mathcal{S}} X(\boldsymbol{\omega} + \boldsymbol{n}\pi) \tag{5}$$

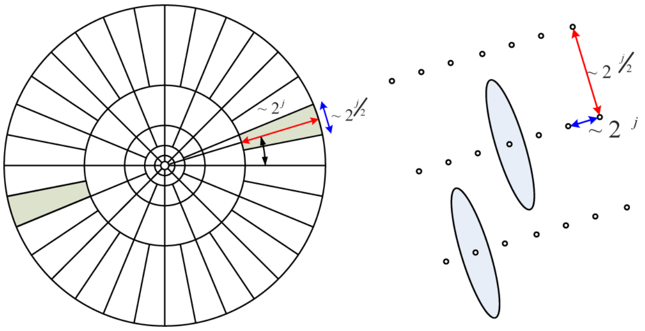
where $\mathcal{S} = \{(0,0),(0,1),(1,0),(1,1)\}$. Visually, $X_{du}(\boldsymbol{\omega})$ is a scaled sum of four copies of $X(\boldsymbol{\omega})$ shifted by $\boldsymbol{n}\pi$, $\boldsymbol{n} \in \mathcal{S}$. Fig. 9(b) and (c) shows the support of windows $u_2(\boldsymbol{\omega})$ and its shifted copies by $\boldsymbol{n}\pi$.

Another result in filter bank theory that will be needed is the equivalent relation between the two structures in Fig. 3. Signal $y(n)$ is obtained from $x(n)$ by decimation by m and filtering by $H(\omega)$. Using (3), it is easy to show that the operation is equivalent to filtering by $H(m\omega)$ followed by decimation by m. This relationship is called "noble property" in signal processing literature.

III. The Curvelet and Contourlet Transforms

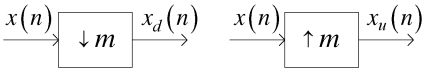
A. The Second Generation Continuous Curvelet Transform

The first generation of curvelet transform is a ridgelet transform of bandpass filtered images [9]. It is not clear what the basis functions are in this construction of curvelet transform. The recent curvelet transform described in [20], sometimes called "second-generation curvelet," is considerably simpler and easier to use. The main idea of this construction is to decompose any



(a) (b)

Fig. 1. Curvelet tiling of space and frequency (a) tiling of the frequency plane, (b) the spatial grid of the curvelet at a given scale and orientation [20].



(a) (b)

Fig. 2. Two basic operations in multirate signal processing: down and up sampling.

2-D function into spaces that are strictly bandpass in frequency domain. The support shapes of these function spaces are concentric wedges [see Fig. 1(a)]. The curvelet coefficients for each scale-direction wedges are estimated as the inner product of the given function and the band-limited curvelet function centered on a grid that is inversely proportional to the wedge-shape support of the curvelet in the frequency domain [see Fig. 1(b)]. Assume that we have two smooth functions W and V, satisfying the "admissibility" condition

$$\sum_{j=-\infty}^{\infty} W^2(2^{-j}r) = 1 \tag{6}$$

$$\sum_{l=-\infty}^{\infty} V^2(t-l) = 1. \tag{7}$$

For each $j > j_0$, the frequency window U_j for the curvelet function is given by

$$U_j(r,\theta) = 2^{-3j/4} W(2^{-j}r) V\left(\frac{2^{\lfloor j/2 \rfloor}\theta}{2\pi}\right). \tag{8}$$

The real-valued curvelet function is defined as a symmetric window function in the frequency domain

$$\hat{\varphi}_j(r,\theta) = U_j(r,\theta) + U_j(r,\theta+\pi). \tag{9}$$

This window is used to define the curvelet function at the first direction of scale j, $\hat{\varphi}_{j,1}(\boldsymbol{\xi}) = \hat{\varphi}_j(r,\theta)$. In the frequency plane, each curvelet direction is indexed by $L = (j,l)$. At each scale j, the number of directions is $2^{\lfloor j/2 \rfloor}$. At scale j, the curvelet direction l is generated by rotating the curvelet window function $\hat{\varphi}_j(r,\theta)$ by a rotation angle $\theta_L = (l-1)\cdot 2\pi \cdot 2^{-\lfloor j/2 \rfloor}$, with $l = 1, 2, \ldots, 2^{\lfloor j/2 \rfloor}$.

The coarse scale curvelet function is defined by polar window $W(r)$ as follows $\hat{\varphi}_0(\boldsymbol{\xi}) = W(r)$. From the definition of $W(r)$ and $V(\theta)$ functions in (6) and (7), it can be shown that the function $\hat{\varphi}_0^2(\boldsymbol{\xi})$ and scaled version of $\hat{\varphi}_{j,l}^2(\boldsymbol{\xi})$ are summed up to one for all $\boldsymbol{\xi}$.

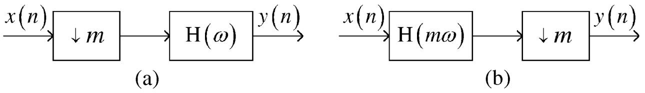


Fig. 3. The two structures are equivalent: the same $x(n)$ will produce the same $y(n)$.

The scale j and angle (or direction) l determine the curvelet in frequency domain, or the shape of the curvelet function. The spatial function also has a translation parameter $\boldsymbol{k} \in \mathbb{Z}^2$. The translation index specifies the amount of translation $t_{\boldsymbol{k}}^{(j,l)}$ that belongs to a grid $\mathcal{M}_{j,l}$. The 2-D curvelet function of variables t_1 and t_2 at scale j and direction θ_L is given by

$$\varphi_{j,l,\boldsymbol{k}} = \varphi_j\left(R_{\theta_L}\left(\boldsymbol{t} - \boldsymbol{t}_{\boldsymbol{k}}^{(j,l)}\right)\right) \tag{10}$$

where R_{θ_L} is rotation by an angle θ_L. The center of the curvelet function is on a rotated grid $\boldsymbol{t}_{\boldsymbol{k}}^{(j,l)} = R_{\theta_L}^{-1}(k_1 2^{-j}, k_2 2^{-j/2})$. An example of the frequency support of a curvelet and its position in the spatial domain is illustrated in Fig. 1.

Let the index parameter μ denote the set of three parameters $(j,l,\boldsymbol{k})$. The curvelet coefficient c_μ is the inner product between an element $f(t_1,t_2) \in L^2(\mathbb{R}^2)$ and a curvelet $\varphi_\mu(\mathbf{t})$

$$c_\mu = \langle f(\boldsymbol{t}), \varphi_\mu(\boldsymbol{t})\rangle = \int_{\mathbb{R}^2} f(\boldsymbol{t})\varphi_\mu(\boldsymbol{t})d\boldsymbol{t}. \tag{11}$$

By definition, the curvelet functions are bandpass functions with compact support in frequency domain. Therefore, the convolution of a two-variable function $f(\boldsymbol{t})$ with a curvelet $\varphi_{j,l}(\boldsymbol{t})$ is also a bandpass function. The curvelet coefficients are obtained by sampling $f(\boldsymbol{t}) * \varphi_{j,l}(\boldsymbol{t})$ on a grid $\mathcal{M}_{j,l}$. If this grid is denser than the Shannon sampling ratio associated with the support in the frequency domain of $\hat{f}(\boldsymbol{\xi})\hat{\varphi}_{j,l}(\boldsymbol{\xi})$, then it is possible to recover the bandpass function from the curvelet coefficients c_μ. Moreover, because the sum of squares of the scale curvelet functions in the frequency domain is equal to 1, the original function can also be recovered:

$$f(\boldsymbol{t}) = \sum_{\mu \in \mathcal{M}} c_\mu \varphi_\mu(\boldsymbol{t}). \tag{12}$$

B. The Fast Discrete Curvelet Transform (FDCT)

Since the theory of the continuous curvelet transform promises an order of magnitude improvement over the wavelet decomposition in many important applications, there have been several implementations of the discrete curvelet transform to operate on digital data.

The FDCT described in [6] with source code (CurveLab package) is available for academic use at www.curvelet.org. This software package actually contains two discrete implementations of the curvelet transform. The first is called digital curvelet transform via unequispaced FFT (USFFT), and the second is called digital curvelet transform via wrapping. In this paper, the FDCT is referred to the wrapping FDCT because

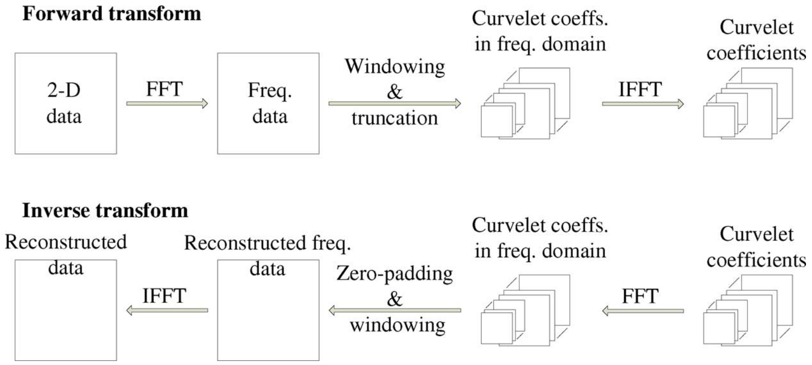


Fig. 4. Data flow structure of the FDCT forward and inverse transforms.

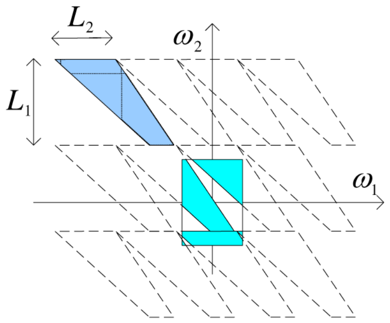


Fig. 5. After multiplication of FFT data with curvelet window, the data on a wedge-shaped support is mapped to a rectangle.

this is the version usually used in actual applications [15], [16], [23].

The wrapping FDCT implementation is based on the FFT algorithm. The data flow diagram of the forward and inverse wrapping FDCT are plotted in Fig. 4. The data are first transformed into the frequency domain by forward FFT. The transformed data are then multiplied with a set of window functions. The shape of these windows are defined according to the requirements of the ideal curvelet transform, such as the parabolic scaling rule. The curvelet coefficients are obtained by inverse FFT from windowing data. Since the window functions are zero except on support regions of elongated wedges, the regions that need to be transformed by the inverse FFT are much smaller than the original data. On the wrapping FDCT, the FFT coefficients on these regions are 'wrapped' or folded into rectangular shape before being applied to inverse FFT algorithm. The size of the rectangle is usually not an integer fraction of the size of original data. This process is equivalent to filtering and subsampling the curvelet subband by rational numbers in two dimensions. Fig. 5 shows the mapping of a wedge-shaped region to several other regions, which lie inside of a rectangle.

The FFT-based FDCT can be viewed as a straightforward discretization of the continuous curvelet transform. Therefore the FDCT is faithful to the definition of its continuous counterpart. However, the FDCT implementation suffers from several practical problems that makes it difficult to use in industrial applications.

1) In the FDCT, the curvelet functions are defined as the product of concentric square functions and sheared angle functions. However, the construction of the FDCT does not take into account specific aspects of discrete transform. For example, the window functions are not automatically 2π periodic. The windows parameters are fixed and have large support areas. As a result, the FDCT has high-redundancy ratio, especially in 3-D.
2) Because of the rational subsampling ratio, the basis of FDCT functions are located on non-integer grids. Moreover, these grids are different for each pair of resolution and direction. This will lead to problems when one wants to exploit the inter-band or inter-scale relationship of curvelet coefficients in actual applications [24].
3) Other inconveniences in implementation of the FDCT are: the curvelet basis functions do not have the same norm; the curvelet coefficients organized into subbands with different size; the redundancy ratio is not fixed, but varies within a range, which may lead to a memory allocation problem.

C. The Contourlet Transform

Parallel to the development of the curvelet transform, there exists another transform proposed in signal processing literature and known as the 'contourlet' transform [8]. If the curvelet transform is originally defined as a transform for continuous functions, the contourlet transform is inherently discrete, without a continuous version. The contourlet transform is implemented by a multirate filter bank; the discrete contourlet basis functions simply being the 2-D filters that are used in that FB.

In the ideal case, the basis functions of the contourlet transform are similar to the basis functions of a discrete curvelet transform. Given the ideal basis functions assumption, the contourlet transform has also the same properties as the curvelet transform. The main differences between the contourlet transform and the FDCT lie in their implementations. The FDCT is faithful to its continuous counterpart. Its basis functions are strictly band-limited in frequency domain. In the spatial domain, they have infinite support. On the other hand, the contourlet transform is constructed from a cascade of finite impulse response discrete filters. Therefore, the calculation of the contourlet transform requires convolution operation. The contourlet bases are finite in the spatial domain. They do not satisfy strict requirements of the curvelet transform such as being a rotation of the same function, or having a finite support region in frequency domain.

In the first construction of the contourlet transform [8], the contourlet functions were not sharply localized in the frequency domain. In a recent construction of the contourlet transform, this problem has been reduced [25], [26]. In the newly designed transform, the actual contourlet functions are quite similar to the curvelet functions in both spatial and frequency domain. Therefore, the two types of discrete transforms should have comparable performances in practical applications.

The contourlet transform, being implemented by a filter bank, is a digital-friendly transform. It has very low redundancy ratio and very fast implementation (faster than the FFT). The problem of this approach is that it is only an approximation of the curvelet transform, because the discrete basis contourlet function does not satisfy the criteria of a curvelet function. According to the construction of the curvelet transform, the basis functions in the Fourier domain need to have a compact support, while the contourlet basis functions, created by cascading digital filters, do not have this property.

IV. 2-D Curvelet Window Definition

Since this section contains rather involved notations, let us make clear our objective. This section aims to construct a family of $2N+1$ smooth 2-D curvelet windows (N is an integer), denoted as $u_l(\boldsymbol{\omega})$, $l = 0, 1, \ldots, 2N$. This family of smooth windows will be used in later sections in defining the curvelet functions in the frequency domain for the UDCT. The set of 2-D functions satisfy the following criteria

- All windows functions are $\mathbf{2}\pi$ periodic in ω_1 and ω_2. In the following, we limit the specification of $u_l(\boldsymbol{\omega})$ to the $(-\pi, \pi)^2$ square.
- The first window $u_0(\boldsymbol{\omega})$ has a square-shape support and zero outside $[-\pi/2, \pi/2]^2$ region [Fig. 9(d)]. This function is sometimes referred to as the lowpass window.
- The other $2N$ windows have wedge-shaped supports [Fig. 9(a)].
- All $u_l(\boldsymbol{\omega})$ are smooth functions, with compact support. The function has value 1 in an "essential support" region and smoothly transition to zero outside a compact support region. The parameter η_a and η_b control the width of these transition regions.
- For the wedge-shaped support function, the widest part of the wedge must be smaller than a fractional of 2π. In the case illustrated in Fig. 9, the widest part of the wedge-shaped support of $u_l(\boldsymbol{\omega})$ windows, $l \neq 0$, should be less than or equal to π.
- The set of $u_0^2(\boldsymbol{\omega})$ and $u_l^2(\boldsymbol{\omega}) + u_l^2(-\boldsymbol{\omega})$ form a partition of unity, which means their sum equal to 1.

The construction of 2-D window functions $u_l(\boldsymbol{\omega})$ are divided in to several steps. We first define a function $\beta(t)$ which has a smooth transition from 0 to 1. Based on the $\beta(t)$ function, a set of 2 smooth concentric square windows and a set of $2N$ angle windows are created. The parameter η_a controls the width of the transition regions of concentric square windows, while parameter η_b controls the transition regions of angle windows. The square-shape support $u_0(\boldsymbol{\omega})$ is created from the product of two 1-D functions. The wedge-shaped supports $u_l(\boldsymbol{\omega})$ are constructed as the products of a concentric square passband window and the set of $2N$ angle windows.

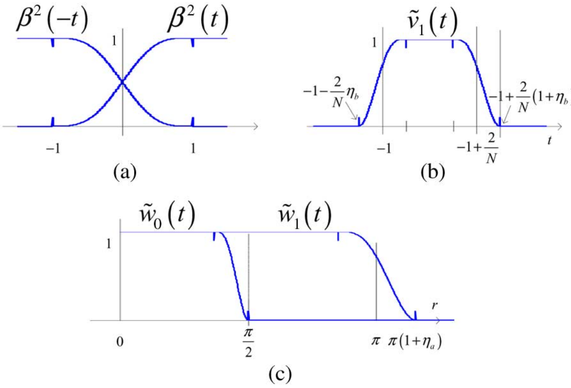


Fig. 6. The functions that are used to define the curvelet functions in frequency domain. (a) $\beta^2(t)$ function (13). (b) $\tilde{v}_1(t)$ function (26). (c) $\tilde{w}_0(t)$ in and $\tilde{w}_1(t)$ functions in (17) and (15).

A. One Dimensional Projector Function

We define a function $\beta(t)$ that has a smooth transition from 0 to 1 when t goes from -1 to 1 [see Fig. 6(a)]. The $\beta(t)$ function satisfies

$$\begin{cases} \beta(t) = 1 & \text{when } 1 \leq t \\ \beta(t) = 0 & \text{when } t \leq -1 \\ \beta^2(t) + \beta^2(-t) = 1 & \text{when } -1 \leq t \leq 1. \end{cases} \tag{13}$$

It is possible to find many smooth functions that satisfy (13). In our implementation we chose the following spline function when $-1 \leq t \leq 1$:

$$\beta^2(t) = -\frac{5}{32}t^7 + \frac{21}{32}t^5 - \frac{35}{32}t^3 + \frac{35}{32}t + \frac{1}{2}. \tag{14}$$

B. Square-Support Functions

Based on the β function, we now define two functions of t, which will be used to define the set of lowpass and concentric bandpass window functions in 2-D plane. The functions $\tilde{w}_1(t)$ are parameterized by $\eta_a < 1/2$

$$\tilde{w}_1(t) = \beta\left(\frac{\pi - |t|}{\pi\eta_a}\right). \tag{15}$$

Essentially, $\tilde{w}_1(t)$ is a window function that has a smooth transition from 1 to 0 when $|t|$ goes from $\pi(1-\eta_a)$ to $\pi(1+\eta_a)$ From definition of $\beta(t)$, it is easy to show that

$$\sum_{n=-\infty}^{\infty} \tilde{w}_1^2(t + 2n\pi) = 1. \tag{16}$$

The function $\tilde{w}_0(t)$ is given as a scaling version of $\tilde{w}_1(t)$, so that the support of $\tilde{w}_0(t)$ is $[-\pi/2, \pi/2]$.

$$\tilde{w}_0(t) = \tilde{w}_1\left(2t(1+\eta_a)\right). \tag{17}$$

The two function $\tilde{w}_0(t)$ and $\tilde{w}_1(t)$ are plotted in Fig. 6(c). A low pass function in the frequency plane is defined from the $\tilde{w}_0(t)$ as follows:

$$w_0(\boldsymbol{\omega}) = \tilde{w}_0(\omega_1)\tilde{w}_0(\omega_2). \tag{18}$$

The $w_0(\boldsymbol{\omega})$ is zero outside the square $[-\pi/2, \pi/2]^2$. We now define $w_1(\boldsymbol{\omega})$ based on lowpass window $w_0(\boldsymbol{\omega})$ and $\tilde{w}_1(t)$ in (15)

$$w_1(\boldsymbol{\omega}) = \left(1 - w_0^2(\boldsymbol{\omega})\right)^{1/2} \tilde{w}_1(\omega_1)\tilde{w}_1(\omega_2). \tag{19}$$

Following the definitions of $w_0(\boldsymbol{\omega})$ and $w_1(\boldsymbol{\omega})$ in (18) and (19)

$$w_0^2(\boldsymbol{\omega}) + w_1^2(\boldsymbol{\omega}) = \tilde{w}_1^2(\omega_1)\tilde{w}_1^2(\omega_2). \tag{20}$$

This is because with $\eta_a < 1/2$, $\tilde{w}_1(\omega_i) = 1$ when $|\omega_i| < \pi/2$ [see Fig. 6(c)]. Therefore, $\tilde{w}_1^2(\omega_1)\tilde{w}_1^2(\omega_2)w_0^2(\boldsymbol{\omega}) = w_0^2(\boldsymbol{\omega})$.

Because our windows will be used as the discrete curvelet functions in frequency domain, they have to be $\mathbf{2}\pi$ periodic. The function $w_0(\boldsymbol{\omega})$ is periodized as follows:

$$u_0(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} w_0(\boldsymbol{\omega} + 2\boldsymbol{n}\pi). \tag{21}$$

A 3-D view of $u_0(\boldsymbol{\omega})$ is in Fig. 9(d). On the $[-\pi, \pi]^2$ square, the 2-D function has a square support $[-\pi/2, \pi/2]^2$. Moreover, by the scaling relationship in (17), $u_0(\boldsymbol{\omega}) = 1$ when $|\omega_1|, |\omega_2| < (\pi/2)((1-\eta_a)/(1+\eta_a))$. If $((1-\eta_a)/(1+\eta_a)) > (1/2)$, we have the following relationship:

$$u_0(2\boldsymbol{\omega})u_0(\boldsymbol{\omega}) = \begin{cases} u_0(2\boldsymbol{\omega}) & \text{when } |\omega_1|, |\omega_2| < \pi/4 \\ 0 & \text{when } \pi/4 \le |\omega_1|, |\omega_2| < \pi. \end{cases} \tag{22}$$

The function $u_0(\boldsymbol{\omega})$ corresponds to a lowpass discrete filter, and the function $w_1(\boldsymbol{\omega})$ is a smooth bandpass function with concentric square support. Intuitively, their sum of squares is equal to one.

$$\begin{aligned} & u_0^2(\boldsymbol{\omega}) + \sum_{\boldsymbol{n}\in\mathbb{Z}^2} w_1^2(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) \\ &= \sum_{\boldsymbol{n}\in\mathbb{Z}^2} w_0^2(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) + w_1^2(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) && (23) \\ &= \sum_{\boldsymbol{n}\in\mathbb{Z}^2} \tilde{w}_1^2(\omega_1 + 2n_1\pi)\tilde{w}_1^2(\omega_2 + 2n_2\pi) && (24) \\ &= 1 && (25) \end{aligned}$$

Equation (23) is obtained from (21) and the fact that $w_0(\boldsymbol{\omega} + 2\boldsymbol{n}\pi)$ are not overlapping. The next two steps are results of (20) and (16).

C. Angle Polar Functions

A set of polar angle functions is defined in this section. Similar to the angle function in continuous case [$V(t)$ in (7)], their squares are summed up to one. However, since the new angle functions are defined for discrete frequency plane, the rotation relationship between two functions will be replaced by the shearing relationship.

Assume that the angle functions that need to be defined is $2N$, with essential support in the range $(-\pi/4, 3\pi/4)$. First, we need to define N intermediary functions $\tilde{v}_l(t)$, $l = 1, \ldots, N$:

$$\tilde{v}_1(t) = \beta\left(\frac{\frac{2}{N} - 1 - t}{\frac{2}{N}\eta_b}\right)\beta\left(\frac{t+1}{\frac{2}{N}\eta_b}\right) \quad \text{with} \quad \eta_b \le \frac{1}{2} \tag{26}$$

$$\tilde{v}_l(t) = \tilde{v}_1\left(t - \frac{2(l-1)}{N}\right) \quad \text{with} \quad l = 2, \ldots, N. \tag{27}$$

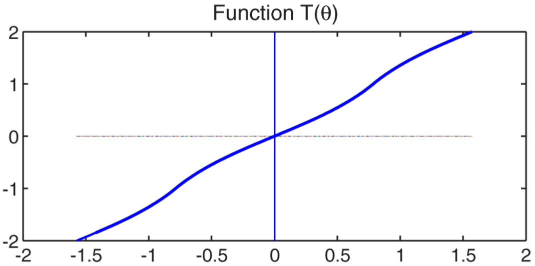


Fig. 7. Function $T(\theta)$ is used to map the 1-D function $\tilde{v}_l(t)$ to polar function $v_l(\theta)$.

The definition of $\tilde{v}_1(t)$ specifies that the function, shown in Fig. 6(b), is a smooth window with main support from -1 to $-1 + 2/N$. The width of the transition area is controlled by parameter η_b; The function changes smoothly from 0 to 1 (or from 1 to 0) when t goes from $-1 - \eta_b 2/N$ to $-1 + \eta_b 2/N$ (or from $-1 + (1-\eta_b)2/N$ to $-1 + (1+\eta_b)2/N$). The condition $\eta_b \le (1/2)$ ensures that the set t satisfying $\tilde{v}_1(t) = 1$ is not empty. The other $\tilde{v}_l(t)$ are shifted copies of $\tilde{v}_1(t)$ by $2(l-1)/N$. From the way $\beta(t)$ was defined in (13), one can verify that the sum of squares of $\tilde{v}_l(t)$, $l = 1, \ldots, N$ is also a smooth window with support from $-1 - 2/N\eta_b$ to $1 + 2/N\eta_b$.

In order to convert $\tilde{v}_l(t)$ to polar functions, we define a function $T(\theta)$ that maps the angle value θ in the range $(-\pi/2, \pi/2)$ to value $(-2, 2)$ as follows:

$$T(\theta) = \begin{cases} \tan(\theta) & \text{when } -\pi/4 \le \theta \le \pi/4 \\ 2 - \tan(\pi/2 - \theta) & \text{when } \pi/4 < \theta < \pi/2 \\ -2 - \tan(\pi/2 - \theta) & \text{when } -\pi/2 < \theta < -\pi/4. \end{cases} \tag{28}$$

From the functions $\tilde{v}_l(t)$, we define N polar angle functions of θ

$$v_l(\theta) = \tilde{v}_l\left(T(\theta)\right). \tag{29}$$

Function $T(\theta)$ is plotted in Fig. 7. This function maps the 1-D functions $v_l(t)$ to polar angle functions $v_l(\theta)$, whose essential support and 3-D views are shown in Fig. 8.

It is easy to check that the two functions $T(\theta \pm \pi/4) \pm 1$ are antisymmetric. This allows us to construct other N polar angle functions by flipping $v_l(\theta)$, $l = 2, \ldots, N-1$ around the value $\pi/4$

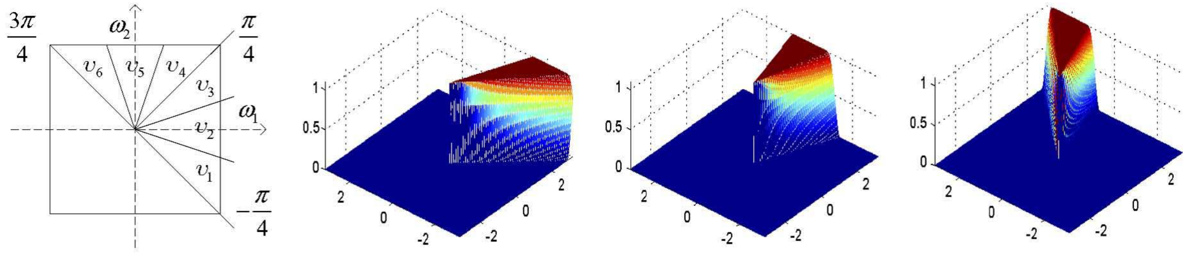
$$v_l(\theta) = v_{2N+1-l}\left(\frac{\pi}{2} - \theta\right), \quad l = N+1, \ldots, 2N. \tag{30}$$

The antisymmetric property of $T(\theta - \pi/4) - 1$ is used to show that the sum of square of $v_l(\theta)$ is still equal to 1 in the overlapping regions of $v_N(\theta)$ and $v_{N+1}(\theta)$. Moreover, the sum of squares of all angle polar functions $v_l(\pm\theta)$ is also equal to one.

$$\sum_{l=1}^{2N} v_l^2(\theta) + v_l^2(-\theta) = 1. \tag{31}$$

D. Discrete Curvelet Windows

In a polar coordinate, $2N$ functions $v_l(\theta)$ are depending only on angle coordinate. Since they are two-variable functions on a



(a) (b) (c) (d)

Fig. 8. (a) Essential supports of the angle functions $v_l(\theta)$ as polar functions, and 3-D view of these polar function in the frequency plane. (b) 2-D angle functions $v_1(\boldsymbol{\omega})$. (c) 2-D angle functions $v_2(\boldsymbol{\omega})$. (d) 2-D angle functions $v_3(\boldsymbol{\omega})$.

plane, we refer to them as function of two variables ω_1 and ω_2, or $v_l(\boldsymbol{\omega})$. Equation (31) is rewritten

$$\sum_{l=1}^{2N} v_l^2(\boldsymbol{\omega}) + v_l^2(-\boldsymbol{\omega}) = 1. \tag{32}$$

We define the following directional wedge functions in the 2-D plane:

$$\tilde{u}_l(\boldsymbol{\omega}) = v_l(\boldsymbol{\omega})w_1(\boldsymbol{\omega}) \quad \text{with} \quad l = 1, 2, \ldots, 2N. \tag{33}$$

From the two above equations and the fact that $w_1(\boldsymbol{\omega})$ is symmetric, it is obvious that

$$\sum_{l=1}^{2N} \tilde{u}_l^2(\boldsymbol{\omega}) + \tilde{u}_l^2(-\boldsymbol{\omega}) = w_1^2(\boldsymbol{\omega}). \tag{34}$$

$2N$ functions $\tilde{u}_l(\boldsymbol{\omega})$ are wedge-shaped support windows with smooth transition regions. In order for a function to correspond to a discrete filter in frequency domain, it has to be $\mathbf{2}\pi$ periodic in both ω_1 and ω_2. A new set of $\mathbf{2}\pi$-periodic functions $u_l(\boldsymbol{\omega})$ is given

$$u_l(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} \tilde{u}_l(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) \tag{35}$$

with $l = 1, 2, \ldots, 2N$. Because $\tilde{u}_l(\boldsymbol{\omega})$ have wedge-shaped compact supports [example of its supports are in Fig. 9(a)], it does not overlap with its shifted copies by $2\boldsymbol{n}\pi$. Thus

$$u_l^2(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} \tilde{u}_l^2(\boldsymbol{\omega} + 2\boldsymbol{n}\pi). \tag{36}$$

Following the relation in (34), we have

$$\sum_{l=1}^{2N} u_l^2(\boldsymbol{\omega}) + u_l^2(-\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} w_1^2(\boldsymbol{\omega} + 2\boldsymbol{n}\pi). \tag{37}$$

Combine with (25), the set of $u_l^2(\boldsymbol{\omega})$ window form a partition of unity

$$u_0^2(\boldsymbol{\omega}) + \sum_{l=1}^{2N} u_l^2(\boldsymbol{\omega}) + u_l^2(-\boldsymbol{\omega}) = 1. \tag{38}$$

The set of $2N+1$ window functions $u_l(\boldsymbol{\omega})$ satisfy all the criteria that have been listed at the beginning of this section.

Example 1: The construction of a set of 2-D windows $u_l(\boldsymbol{\omega})$ with three parameters $2N = 6$, $\eta_a = \eta_b = 0.15$ is illustrated in Figs. 8 and 9.

V. The Uniform Discrete Curvelet Transform as a Filter Bank

The uniform discrete curvelet transform as a FB is defined in this section, using the parameterized family of smooth windowed functions $u_l^2(\omega_1, \omega_2)$, $l = 0, 1, \ldots, 2N$. The UDCT is built up as a simple 2-D FB with one lowpass band and six directional highpass bands. Then it is shown that by cascading the same FB at lower resolution and fixing the number of directional bands following the parabolic scaling rule, a discrete decomposition faithful to the definition of the curvelet transform is created.

A. 2-D Filter Bank Implementation in the Frequency Domain

1) A 2-D Filter Bank Using the Set of 7 Windows Defined in Example 1: In example 1, we have defined a parameterized family of 2-D windows that forms a partition of unity [(38)]. Let us define a 7-band FB with the analysis filters $F_l(\boldsymbol{\omega})$ defined from 7 $u_l(\boldsymbol{\omega})$ windows as follows:

$$F_0(\boldsymbol{\omega}) = 2u_0(\boldsymbol{\omega}) \tag{39}$$

$$F_l(\boldsymbol{\omega}) = 2\sqrt{2}u_l(\boldsymbol{\omega}), \quad l = 1, \ldots, 6 \tag{40}$$

$$G_l(\boldsymbol{\omega}) = F_l(\boldsymbol{\omega}). \tag{41}$$

In the spatial domain, the synthesis filters $g_l(\boldsymbol{n})$ are the same as the analysis filters, $g_l(\boldsymbol{n}) = f_l(\boldsymbol{n})$. The decimation ratio for all subbands is $2\boldsymbol{I}$. At the output of the FB, only real values of the reconstructed image are kept. In the following, we will show that the FB illustrated in Fig. 10 implements a discrete transform, or the FB is perfect reconstruction.

Let us consider the decomposition and reconstruction of the FB for a 2-D signal $x(\boldsymbol{n})$. The signal $x(\boldsymbol{n})$ is first filtered by $f_l(\boldsymbol{n})$. The filtered signals are decimated by $2\boldsymbol{I}$ (keeping every other row and column). On the synthesis side, the subband images are first upsampled by $2\boldsymbol{I}$ (alternating every row and column with zero), then convolved with $g_l(\boldsymbol{n})$. By the multirate filter bank theory [21], these signals can be written in the frequency domain

$$X_l(\boldsymbol{\omega}) = \frac{1}{4}G_l(\boldsymbol{\omega})\sum_{\boldsymbol{n}\in\mathcal{S}} X(\boldsymbol{\omega} + \boldsymbol{n}\pi)F_l(\boldsymbol{\omega} + \boldsymbol{n}\pi) \tag{42}$$

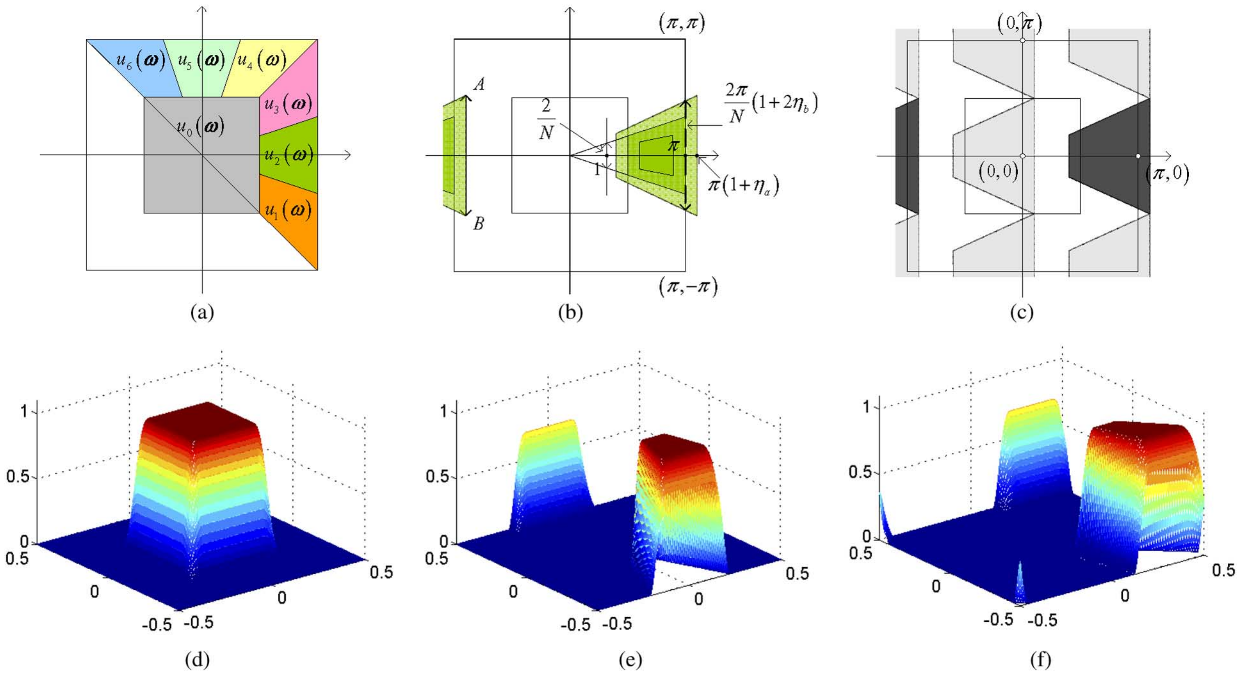


Fig. 9. An example of a family of 7 $u_l(\boldsymbol{\omega})$ windows in (ω_1, ω_2) plane. (a) Regions of essential support of $u_l(\boldsymbol{\omega})$. (b) Support of $u_2(\boldsymbol{\omega})$, distance AB can be estimated from N, η_a and η_b. (c) The support area of $F_2(\boldsymbol{\omega})$ and $G_2(\boldsymbol{\omega})$ (darker area) that are built from function $u_2(\boldsymbol{\omega})$. The lighter shade areas are support regions of $F_l(\boldsymbol{\omega} + \boldsymbol{n}\pi)$, $n \in \{\mathcal{S} \setminus (0,0)\}$. (d), (e), (f) 3-D view of functions $u_0(\boldsymbol{\omega})$, $u_2(\boldsymbol{\omega})$, $u_3(\boldsymbol{\omega})$.

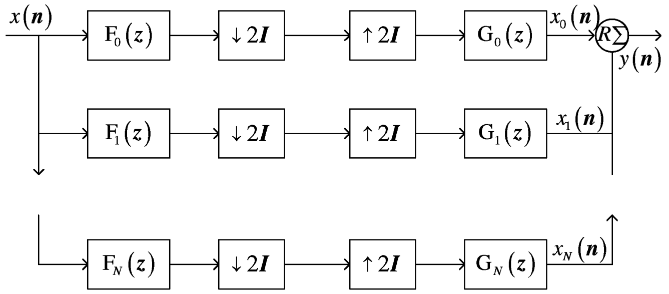

$x(\boldsymbol{n})$
$F_0(z)$
$\downarrow 2\boldsymbol{I}$
$\uparrow 2\boldsymbol{I}$
$G_0(z)$
$x_0(\boldsymbol{n})$
$R\Sigma$
$y(\boldsymbol{n})$
$F_1(z)$
$\downarrow 2\boldsymbol{I}$
$\uparrow 2\boldsymbol{I}$
$G_1(z)$
$x_1(\boldsymbol{n})$
$F_N(z)$
$\downarrow 2\boldsymbol{I}$
$\uparrow 2\boldsymbol{I}$
$G_N(z)$
$x_N(\boldsymbol{n})$

Fig. 10. A 7-band FB constructed from $u_l(\boldsymbol{\omega})$ windows in Example 1.

where $\mathcal{S} = \{(0,0),(0,1),(1,0),(1,1)\}$. Finally, the reconstructed output signal is obtained by $y(\boldsymbol{n}) = \mathrm{Real}(\sum_{l=0}^{2N}(x_l(\boldsymbol{n}))$.

The wedge-shaped support of window function $u_2(\boldsymbol{\omega})$ (or the filter $F_2(\boldsymbol{\omega})$) in Fig. 9(b) has the widest part $AB = (2\pi/N)(1+2\eta_b)(1+\eta_a)$. Because $\eta_a = \eta_b = 0.15$ and $N = 3$, the distance AB is less than π. Therefore, the support of $F_l(\boldsymbol{\omega}+\boldsymbol{n}\pi), \boldsymbol{n} \in \mathbb{Z}^2$, are not overlapping. Therefore

$$G_l(\boldsymbol{\omega}) \sum_{\boldsymbol{n}\in\mathcal{S}} F_l(\boldsymbol{\omega} + \boldsymbol{n}\pi) = G_l(\boldsymbol{\omega})F_l(\boldsymbol{\omega}). \tag{43}$$

Equation (42) is rewritten

$$X_l(\boldsymbol{\omega}) = \frac{1}{4}X(\boldsymbol{\omega})F_l(\boldsymbol{\omega})G_l(\boldsymbol{\omega}). \tag{44}$$

Since the output $y(\boldsymbol{n})$ is obtained from the real part of a complex signal, its Fourier transform is given

$$Y(\boldsymbol{\omega}) = \frac{1}{2}\sum_{l=0}^{6}\left(X_l(\boldsymbol{\omega}) + X_l^*(-\boldsymbol{\omega})\right). \tag{45}$$

Combining (44) and (45) and note that $X(\boldsymbol{\omega}) = X^*(-\boldsymbol{\omega})$, we have

$$Y(\boldsymbol{\omega}) = \frac{1}{8}X(\boldsymbol{\omega})\sum_{l=0}^{6}\left(G_l(\boldsymbol{\omega})F_l(\boldsymbol{\omega}) + G_l^*(-\boldsymbol{\omega})F_l^*(-\boldsymbol{\omega})\right). \tag{46}$$

Since $F_l(\boldsymbol{\omega})$ and $G_l(\boldsymbol{\omega})$ are defined from $u_l(\boldsymbol{\omega})$, both are positive functions. Therefore, we have $G_l(\boldsymbol{\omega})F_l(\boldsymbol{\omega}) = 8u_l^2(\boldsymbol{\omega})$, $l = 1,\ldots,6$. The lowpass function $u_0(\boldsymbol{\omega})$ is symmetric, which leads to $G_0(\boldsymbol{\omega})F_0(\boldsymbol{\omega}) = G_0^*(-\boldsymbol{\omega})F_0^*(-\boldsymbol{\omega}) = 4u_0^2(\boldsymbol{\omega})$. Equation (46) is rewritten

$$Y(\boldsymbol{\omega}) = X(\boldsymbol{\omega})\left(u_0^2(\boldsymbol{\omega}) + \sum_{l=1}^{6} u_l^2(\boldsymbol{\omega}) + u_l^2(-\boldsymbol{\omega})\right). \tag{47}$$

From the (38), we have $Y(\boldsymbol{\omega}) = X(\boldsymbol{\omega})$, or $y(\boldsymbol{n}) = x(\boldsymbol{n})$.

2) Relationship Between η_a, η_b and the Redundancy of the Decomposition: In order to reduce the overcomplete ratio of a discrete curvelet decomposition, one would need to decimate the subband after filtering. We limit our choice of decimation ratio to the power of 2. The shape of support of a function $u_l(\omega_1,\omega_2)$ is fully determined by parameters η_a, η_b and N. The distance AB in Fig. 9 is $(2\pi/N)(1+2\eta_b)(1+\eta_a)$. Our objective is to determine η_a, and η_b so that the distance $AB < \pi/2^n$. Following are set of parameters that satisfy this requirement:

$$N = 3\cdot 2^n, \quad n \geq 0, \quad \eta_a = \eta_b = 0.15 \tag{48}$$

$$N = 4\cdot 2^n, \quad n \geq 0, \quad \eta_a = \eta_b = 0.3 \tag{49}$$

$$N = 5\cdot 2^n, \quad n \geq 0, \quad \eta_a = \eta_b = 0.5. \tag{50}$$

From now on we pick the parameters $\eta_a = \eta_b = 0.15$ and N in the form $3\cdot 2^n$ as in (48). This set of parameters is interesting because it allows the curvelet window have a regions of

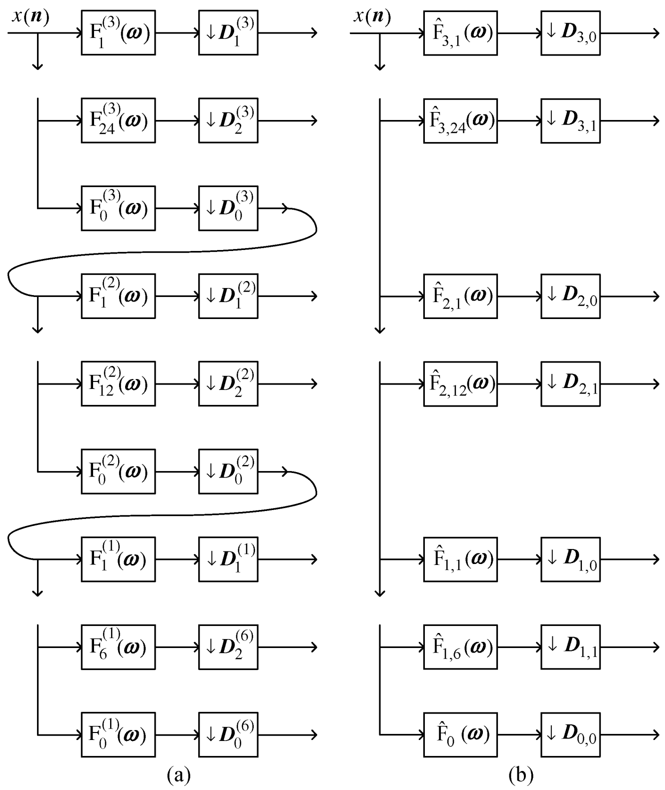

$x(\boldsymbol{n})$
$F_1^{(3)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_1^{(3)}$
$F_{24}^{(3)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_2^{(3)}$
$F_0^{(3)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_0^{(3)}$
$F_1^{(2)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_1^{(2)}$
$F_{12}^{(2)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_2^{(2)}$
$F_0^{(2)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_0^{(2)}$
$F_1^{(1)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_1^{(1)}$
$F_6^{(1)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_2^{(6)}$
$F_0^{(1)}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_0^{(6)}$
(a)
$x(\boldsymbol{n})$
$\hat{F}_{3,1}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{3,0}$
$\hat{F}_{3,24}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{3,1}$
$\hat{F}_{2,1}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{2,0}$
$\hat{F}_{2,12}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{2,1}$
$\hat{F}_{1,1}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{1,0}$
$\hat{F}_{1,6}(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{1,1}$
$\hat{F}_0(\boldsymbol{\omega})$
$\downarrow \boldsymbol{D}_{0,0}$
(b)

Fig. 11. Discrete curvelet transform is obtained by cascading multiple level of UDCT filter bank. (a) The three-level UDCT filter bank in Example 2. (b) Equivalent FB, filters $\hat{F}_{j,l}(\boldsymbol{\omega})$ as in (55), and $\boldsymbol{D}_{j,0}$, $\boldsymbol{D}_{j,1}$, as in (59) and (60).

transition around the essential support region of the curvelet. It makes the curvelet well localized in the spatial domain. Moreover, the estimation of the redundancy ratio in (86) shows that the 2-D UDCT with this set of parameters has an acceptable redundancy of 4.

3) Two-Dimensional Filter Bank With $2N$ Directional Band: We have created a 7-band FB based on the set of 7 $u_l(\boldsymbol{\omega})$ windows. In general case, we set $N = 3 \cdot 2^n$, $n \geq 0$, $\eta_a = \eta_b = 0.15$ as in (48) and use the $2N+1$ $u_l(\boldsymbol{\omega})$ windows to define a $(2N+1)$-band FB as follows:

$$F_0(\boldsymbol{\omega}) = 2u_0(\boldsymbol{\omega}) \tag{51}$$

$$F_l(\boldsymbol{\omega}) = 2^{\frac{n+3}{2}} u_l(\boldsymbol{\omega}), \quad l = 1, \dots, 2N \tag{52}$$

$$G_l(\boldsymbol{\omega}) = F_l(\boldsymbol{\omega}). \tag{53}$$

Since the two parameters $\eta_a = \eta_b = 0.15$, it can be shown that the widest part of the wedge-shaped supports of $u_l(\boldsymbol{\omega})$ windows are less than $\pi/(N/3)$, or $\pi/2^n$. Therefore, the decimation ratio for the $2N$ directional bands of the $2N$-band FB will be 2^n times higher than the case $2N = 6$. The decimation ratio for the first $3 \cdot 2^n$ directional bands are $\boldsymbol{D}_1^{(N)} = \operatorname{diag}\{2, 2^{n+1}\}$ and for the last $3 \cdot 2^n$ directional bands are $\boldsymbol{D}_2^{(N)} = \operatorname{diag}\{2^{n+1}, 2\}$. The decimation ratio for the lowpass band is $\boldsymbol{D}_0^{(N)} = 2\boldsymbol{I}$. Let us rewrite the three decimation ratios used in a $2N$-band UDCT filter bank as follows:

$$\boldsymbol{D}_0^{(N)} = 2\boldsymbol{I}, \quad \boldsymbol{D}_1^{(N)} = \operatorname{diag}\left\{2, \frac{2N}{3}\right\}, \quad \boldsymbol{D}_2^{(N)} = \operatorname{diag}\left\{\frac{2N}{3}, 2\right\}. \tag{54}$$

Following similar argument as for the case of 7-band FB, it is straightforward to show that the constructed FB is indeed perfect reconstruction. Since the synthesis filters are the same as the analysis filters, and there is no aliasing in the subsampling subbands, the norm of the output subbands is the same as the norm of input signal; The FB is implementing a norm-preserving (or tight frame) and self-dual transform [27].

B. The Uniform Discrete Curvelet Transform as an Iterative Multiscale Filter Bank

We have constructed a set of $2N$ 2-D directional filters $F_l(\boldsymbol{\omega})$, $l = 1, \dots, 2N$ and a lowpass filter $F_0(\boldsymbol{\omega})$ in such a way that the directional subbands and the lowpass subband can be decimated without aliasing. The defined filters in the frequency domain are real-valued functions and satisfy the perfect reconstruction condition, taking into account the decimation ratio. Similar to the complex wavelet [4], our directional filters have one-sided support in the frequency domain, making the subband coefficients complex valued. In the reconstruction procedure, the final complex components are simply discarded.

In order to obtain a multiresolution decomposition, another FB with the same η_a, η_b is reiterated at lower band. Since our objective is to create a discrete decomposition faithful to the curvelet transform, the number of directional subbands should follow the parabolic scaling rule. For example, a signal $x(\boldsymbol{n})$ can be decomposed by the UDCT into one lowpass band and J highpass bands, numbered from 1 (lowest resolution) to J (highest resolution). The UDCT is implemented by cascading J multiresolution FBs that are constructed in the previous subsection. The number of directional bands $2N_j$ at resolution j should be proportional to $2^{\lfloor j/2 \rfloor}$.

However, in a practical implementation, the number of directions of UDCT filter bank at each level is flexible. For instance, the multiresolution FB consist of J level; the UDCT filter bank at level j has $2N_j$ directions. The three parameters η_a, η_b and $2N_j$ are chosen as in (48). The filters at scale j are denoted by $F_l^{(j)}(\boldsymbol{\omega})$, $l = 0, 1, \dots, 2N_j$. Since we use the same parameter η_a for all J set of parameterized windows, the lowpass windows $F_0^{(j)}(\boldsymbol{\omega})$ are the same for all scale j.

Example 2: A Multilevel UDCT Decomposition: Let us demonstrate an example of the UDCT decomposition, with configuration as follows: $J = 3$, $2N_1 = 6$, $2N_2 = 12$, $2N_3 = 24$. The 3 level UDCT FB associated with this decomposition is in Fig. 11(a).

Fig. 12 are examples of the zoneplate image decomposed by our discrete curvelet transform with above parameters J and N_j. Fig. 12(c) is the magnitude of the complex value coefficients in the transform domain.

For any multilevel FB, one can find an equivalent one-level FB, by moving the decimation block towards the end (or the start) of the analysis-side (or synthesis-side) FB. The equivalent FB for structure in Fig. 11(a) is in Fig. 11(b), with equivalent filters in scale j and direction l denoted by $\hat{F}_{j,l}(\boldsymbol{\omega})$. By the filter bank theory [21]

$$\hat{F}_{j,l}(\boldsymbol{\omega}) = F_l^{(j)}(2^{J-j}\boldsymbol{\omega}) \prod_{i=0}^{J-j-1} F_0^{(J-i)}(2^i \boldsymbol{\omega}). \tag{55}$$

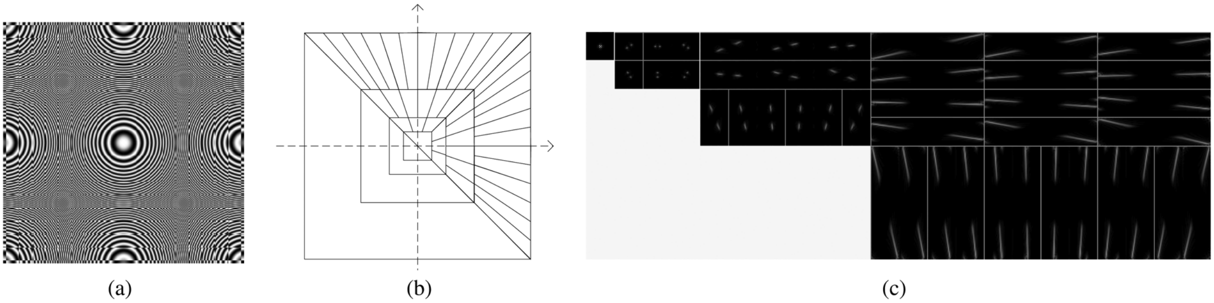


Fig. 12. An example of the UDCT decomposition (a) zoneplate image; (b) essential support of the curvelet function in the Fourier domain; and (c) curvelet coefficients.

Since all $F_0^{(i)}(\boldsymbol{\omega})$, $i = 1, \ldots, J$ are the same and defined from $u_0(\boldsymbol{\omega})$ as in (51), and the product $u_0(\boldsymbol{\omega})u_0(2\boldsymbol{\omega})$ is already estimated in (22), we have

$$\begin{aligned}&\prod_{i=0}^{j} F_0^{(J-i)}(2^i\boldsymbol{\omega})\\ &= \begin{cases} 2^j F_0^{(J-j)}(2^j\boldsymbol{\omega}), & \text{when } |\omega_1|, |\omega_2| < \pi/2^{j+1} \\ 0, & \text{when } \pi/2^{j+1} \le |\omega_1|, |\omega_2| < \pi. \end{cases}\end{aligned} \quad (56)$$

Therefore, in the $[-\pi, \pi]^2$ square of the $\boldsymbol{\omega}$ frequency plane, $\hat{F}_{j,l}(\boldsymbol{\omega})$ when $j < J$ has the following value:

$$\hat{F}_{j,l}(\boldsymbol{\omega}) = \begin{cases} 2^{J-j-1} F_l^{(j)}(2^{J-j}\boldsymbol{\omega}) F_0^{(j+1)}(2^{J-j-1}\boldsymbol{\omega}), \\ \qquad \text{when } |\omega_1|, |\omega_2| < \pi/2^{J-j} \\ 0, \quad \text{when } \pi/2^{J-j} \le |\omega_1|, |\omega_2| < \pi \end{cases} \quad (57)$$

where $F_0(\boldsymbol{\omega})$ is defined in (39) and (21). The equivalent lowpass filter is

$$\hat{F}_0(\boldsymbol{\omega}) = \begin{cases} 2^{J-1} F_0(2^{J-1}\boldsymbol{\omega}) & \text{when } |\omega_1|, |\omega_2| < \pi/2^J \\ 0 & \text{when } \pi/2^J \le |\omega_1|, |\omega_2| < \pi. \end{cases} \quad (58)$$

We have determined the equivalent curvelet filters of an iterated UDCT filter bank in (57) and (58) by pushing all the analysis filters to the left and all the decimation blocks to the right on the analysis side. Associated with the equivalent filters are the overall decimation ratio, which are the product of all decimation blocks along the branch of the iterative FB. The decimation ratio for the first half of directional band (mostly horizontal band) is

$$\boldsymbol{D}_{j,0} = \boldsymbol{D}_1^{(j)} \prod_{i=j+1}^{J} \boldsymbol{D}_0^{(i)} = \operatorname{diag}\left\{2, \frac{2N_j}{3}\right\} \cdot 2^{J-j} \quad (59)$$

and for the second half of directional band (mostly vertical band) is

$$\boldsymbol{D}_{j,1} = \boldsymbol{D}_2^{(j)} \prod_{i=j+1}^{J} \boldsymbol{D}_0^{(i)} = \operatorname{diag}\left\{\frac{2N_j}{3}, 2\right\} \cdot 2^{J-j}. \quad (60)$$

The two above equations are obtained simply by applying (54). The decimation ratio corresponds to the lowpass window $\hat{F}_0(\boldsymbol{\omega})$ in (58) is $\boldsymbol{D}_{0,0} = \operatorname{diag}\{2^J, 2^J\}$.

Example 2 (continued). In the case of example 2, the equivalent FB in Fig. 11(b) produces the same decomposition as the 3 level FB in Fig. 11(a). The overall decimation ratio for the equivalent FB are

$$\begin{aligned} \boldsymbol{D}_{0,0} &= \operatorname{diag}\{8, 8\}, \\ \boldsymbol{D}_{1,0} &= \operatorname{diag}\{8, 8\}, \quad \boldsymbol{D}_{1,1} = \operatorname{diag}\{8, 8\} \\ \boldsymbol{D}_{2,0} &= \operatorname{diag}\{4, 8\}, \quad \boldsymbol{D}_{2,1} = \operatorname{diag}\{8, 4\} \\ \boldsymbol{D}_{3,0} &= \operatorname{diag}\{2, 8\}, \quad \boldsymbol{D}_{3,1} = \operatorname{diag}\{8, 2\}. \end{aligned} \quad (61)$$

VI. The Uniform Discrete Curvelet Transform

A. The Discrete Transform Implemented by UDCT Filter Bank

The UDCT has been introduced as a multiresolution FB implemented in the frequency domain by using a set of curvelet windows. The curvelet basis functions of the UDCT are estimated from the FB implementation as in (57) and (58). Since the directional filters used in the construction of UDCT have one-sided support in the frequency domain, they have complex valued coefficients in the spatial domain. Therefore we have to interpret each complex valued subband coefficient produced by UDCT as the projection of input signal onto two curvelet basis functions, which are the real and imaginary parts of a complex function. The first function is symmetric and the second function is anti-symmetric. Their centers of symmetry and antisymmetry are at the same point, which is also considered as the position of the two curvelet functions.

The index set for the discrete basis function is $\mu = (i, j, l, \boldsymbol{k}) \in \mathcal{M}$, where $i = 1, 2$ is an indicator for the symmetric/antisymmetric property of the basis function; parameter $j = 1, \ldots, J$ is the scale index, $l = 1, \ldots, 2N_j$ is the directional index, and $\boldsymbol{k}$ belongs to the lattice $\Lambda(\boldsymbol{D}_{j,\lfloor (l-1)/N_j \rfloor})$. The frequency domain representation of $\hat{F}_{j,l}(\boldsymbol{\omega})$ in (57) corresponds to a complex function in the spatial domain whose real part is $\varphi_{1,j,l,\mathbf{0}}(\boldsymbol{n})$ and imaginary part is $\varphi_{2,j,l,\mathbf{0}}(\boldsymbol{n})$. The other functions at different location indices $\boldsymbol{k}$ are simply shifted versions of this function on the $\Lambda(\boldsymbol{D}_{j,\lfloor (l-1)/N_j \rfloor})$ grid.

$$\varphi_{(i,j,l,\boldsymbol{k})}(\boldsymbol{n}) = \varphi_{(i,j,l,\mathbf{0})}(\boldsymbol{n} - \boldsymbol{k}) \quad (62)$$

We extend the index set $\mathcal{M}$ to include index elements $\mu = (1, 0, 0, \boldsymbol{k})$ for the "scaling" curvelet that corresponds to the lowpass filter defined in (58). The Fourier transform of $\varphi_{(1,0,0,\mathbf{0})}(\boldsymbol{n})$

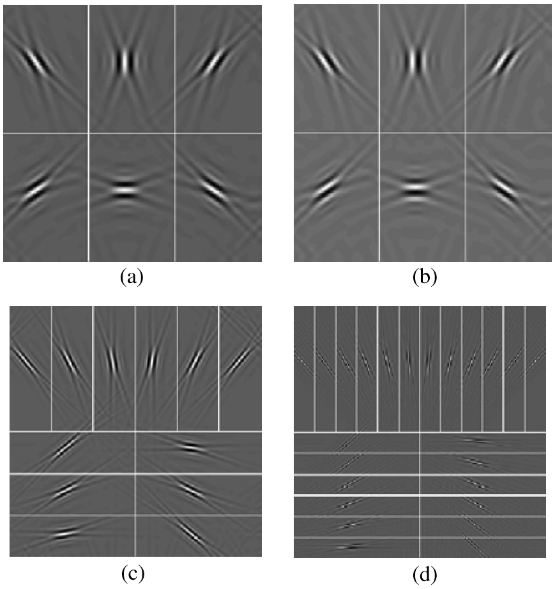


Fig. 13. Examples of 2-D UDCT basis functions corresponds to a 6, 12, 24 configuration. (a) $\varphi_{(1,1,l,\boldsymbol{k})}(\boldsymbol{n}), l = 1,\ldots,2N_1 = 6$. (b) $\varphi_{(2,1,l,\boldsymbol{k})}(\boldsymbol{n}), l = 1,\ldots,2N_1 = 6$. (c) $\varphi_{(1,2,l,\boldsymbol{k})}(\boldsymbol{n}), l = 1,\ldots,2N_2 = 12$. (d) $\varphi_{(1,3,l,\boldsymbol{k})}(\boldsymbol{n}), l = 1,\ldots,2N_3 = 24$.

is $\hat{F}_0(\boldsymbol{\omega})$. The other elements of these lowpass curvelets correspond to $\mathbf{k} \in \Lambda(\boldsymbol{D}_{0,0})$ lattice.

Example 2 (continued). The decomposition in this case has three level of directional curvelet. The lowest directional scale has six direction, where the corresponding symmetric and antisymmetric curvelet functions are shown in Fig. 13(a) and (b). The center of these curvelet functions are situated on the lattices $\Lambda(\boldsymbol{D}_{1,0})$ and $\Lambda(\boldsymbol{D}_{1,1})$ in Fig. 14(a) and (b). Other lattices for scales $J = 2, 3$ are also included in Fig. 14.

Any two dimensional discrete function $x(\boldsymbol{n}) \in \ell^2$ is expressed as a sum of φ_μ.

$$x(\boldsymbol{n}) = \sum_{\mu \in \mathcal{M}} c_\mu \varphi_\mu(\boldsymbol{n}) \tag{63}$$

where c_μ are estimated by

$$c_\mu = \langle x(\boldsymbol{n}), \varphi_\mu(\boldsymbol{n}) \rangle \tag{64}$$

c_μ are the real and imaginary part of complex coefficients produced by the UDCT filter bank. Combining (63) and (64) and writing out the range of index in $\mathcal{M}$, we have the following self-inversion formula:

$$x(\boldsymbol{n}) = \sum_{\boldsymbol{k}\in\Lambda(\boldsymbol{D}_{0,0})} \left\langle x(\boldsymbol{n}), \varphi_{(1,0,0,\boldsymbol{k})}(\boldsymbol{n}) \right\rangle \varphi_{(1,0,0,\boldsymbol{k})}(\boldsymbol{n}) + \sum_{j=1}^{J}\sum_{l=1}^{2N_j}\sum_{i=1}^{2} \sum_{\boldsymbol{k}\in\Lambda\left(\boldsymbol{D}_{j,\left\lfloor\frac{l-1}{N_j}\right\rfloor}\right)} \left\langle x(\boldsymbol{n}), \varphi_{(i,j,l,\boldsymbol{k})}(\boldsymbol{n}) \right\rangle \varphi_{(i,j,l,\boldsymbol{k})}(\boldsymbol{n}). \tag{65}$$

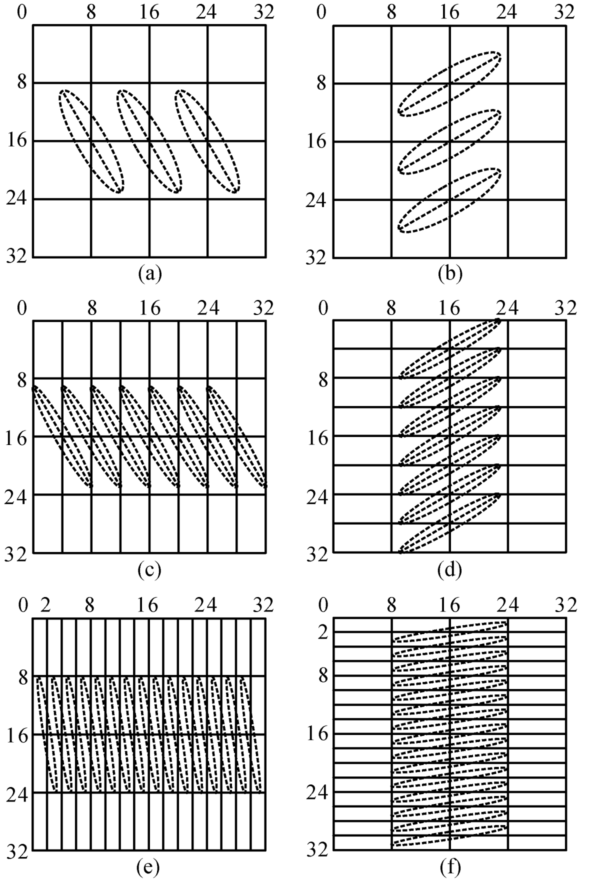


Fig. 14. The lattices of UDCT, corresponding to the UDCT described in Example 2. (a) and (b), (c) and (d), (e) and (f) are lattices for the first, second and third directional resolutions, see (61). (a) $\Lambda(\boldsymbol{D}_{1,0})$. (b) $\Lambda(\boldsymbol{D}_{1,1})$. (c) $\Lambda(\boldsymbol{D}_{2,0})$. (d) $\Lambda(\boldsymbol{D}_{2,1})$. (e) $\Lambda(\boldsymbol{D}_{3,0})$. (f) $\Lambda(\boldsymbol{D}_{3,1})$.

B. The UDCT and Other Curvelet Transforms

The UDCT has some similarities to the wrapping-based FDCT [6], in the sense that both the two transforms employ alias free subsampling in frequency domain. The curvelet functions of the two discrete transforms have compact support windows in Fourier domain. Both the UDCT and the FDCT are implemented by FFT algorithm. Therefore they have the same data flow structure as illustrated in Fig. 4. One can regard the UDCT as a generalization and parametrization of the FDCT. The practical advantages of the UDCT are detailed in the following list, comparing against the list in Section III-B

1) In the UDCT, the curvelet window functions are systematically defined and parameterized by η_a and η_b. These parameters measure the trade off between redundancy ratio and effective support of curvelet functions in the spatial domain. By varying these parameters, it is possible generate UDCT with different redundancy ratio and curvelet support shapes.
2) The UDCT is built up using multirate FB theory. The set of window functions is generated from a cascade of multiresolution FBs, each FB is implemented in frequency domain. This makes all the differences between the UDCT and the FDCT. For example, because the FB employed two integer subsampling matrices for all the directional bands, the generated curvelet functions are located on two lattices; one for mostly vertical and one for mostly horizontal curvelet. The discrete curvelet functions at each resolution and scale are the same function shifted to different nodes on one lattice.

Moreover, the lattices of lower scales are subset of those at higher scales. Therefore, the UDCT coefficients have a clear parent-children relationship.

3) The UDCT also provide several other properties that should be useful in practical applications: i) All the norms of the curvelet functions at each scale are approximately the same. ii) The number of coefficients at each scale are fixed and do not depend on the number of directions. iii) The size of subbands are the same for each scale (subject to a transpose operation).

The UDCT can also be viewed as a complex wavelet decomposition, implemented by filter bank, in the same way as shiftable complex directional pyramid [19] or other complex wavelet transforms [28]. Unlike the contourlet FB, each complex band of the UDCT is shift invariant in the energy sense [29], and should offer better performances in image analysis applications. It is possible to modify the contourlet FB to achieve complete linear shift invariant as in the nonsubsampled contourlet transform [18].

VII. The UDCT in Three or More Dimensions

One of the advantages of the UDCT is that it can be generalized to M-dimensions without significant difficulty. The construction of the M-D UDCT follows the same steps as the 2-D UDCT. First a parameterized family of M variable window functions are defined. These window functions are used to define M-D filters for a M-D FB. By cascading iteratively this M-D FB, a M-D UDCT decomposition is obtained. We focus on how to construct a parameterized family of M-D curvelet windows. The details on using this set of windows to construct a M-D FB, and cascading multiple M-D FBs to create M-D UDCT are omitted. The 3-D UDCT case is treated as an example of the M-D cases.

A. A Set of M-D Curvelet Window Functions

In this section, functions with bold letters denote M-D function. The M-D curvelet windows are defined as the product of a concentric square window and a set of angle windows in M-D space. The M-D generalization of a concentric square window is straightforward. We reuse the two functions $\tilde{w}_0(t)$ and $\tilde{w}_1(t)$ defined in (15) and (17). Similar to the 2-D case in (18) and (19), the two functions $w_0(\boldsymbol{\omega})$ and $w_1(\boldsymbol{\omega})$ in M-D space are given

$$w_0(\boldsymbol{\omega}) = \prod_{k=1}^{M} \tilde{w}_0(\omega_k) \tag{66}$$

$$w_1(\boldsymbol{\omega}) = \left(1 - w_0^2(\boldsymbol{\omega})\right)^{1/2} \prod_{k=1}^{M} \tilde{w}_1(\omega_k). \tag{67}$$

Let us consider the generalization of angle functions in M dimensions. We begin by assuming that we have a set of polar angle functions $v_l(\theta)$, $l = 1\ldots, N$, $N = 3 \cdot 2^n$ as defined in Section IV-C. The main support of the sum of square of these functions is an infinite pyramid shape region satisfying $\omega_1 > |\omega_2|$; On a polar coordinate, its values depend only on the angle coordinate. Three functions $v_1(\theta)$, $v_2(\theta)$ and $v_3(\theta)$ are illustrated in Fig. 8(b), (c), (d), in the case $N = 3$. Instead of considering v_l as a function of angle θ, we write it as a function of (ω_1, ω_2). Building up from 2-D angle function $v_l(\omega_1, \omega_2)$, we define a set of $M \times N^{M-1}$ polar functions on M-D space $\tilde{v}_{\boldsymbol{l}}(\boldsymbol{\omega})$, indexed by $\boldsymbol{l} = (l_1, \ldots, l_M)$, where l_1 is the index of the hyperpyramid, $1 \leq l_1 \leq M$; the rest are $M-1$ direction indices of the angle function within that hyperpyramid, $1 \leq l_2, \ldots, l_M \leq N$. Each function is defined from $M-1$ functions v_l as follows:

$$\begin{aligned}\tilde{v}_{l_1\ldots l_M}(\omega_1, \ldots, \omega_M) \\ = v_{l_2}(\omega_{l_1}, \omega_{k_1}) v_{l_3}(\omega_{l_1}, \omega_{k_2}) \ldots v_{l_M}\left(\omega_{l_1}, \omega_{k_{M-1}}\right)\end{aligned} \tag{68}$$

where $1 \leq k_1 < k_2 \ldots < k_{M-1} \leq M$ and all $k_m \neq l_1$, $1 \leq m \leq M-1$.

For each l_1, N^{M-1} function $\tilde{v}_{l_1,\ldots,l_M}(\pm\boldsymbol{\omega})$ covers a hyper pyramid in l_1 direction of M dimensional space. Unlike the 2-D case, the sum of square of these functions and their mirror through the origin are not equal to 1 in the regions where three or more $\tilde{v}_{\boldsymbol{l}}(\pm\boldsymbol{\omega})$ functions overlap. Denote their sum of square as

$$V^2(\boldsymbol{\omega}) = \sum_{l_1=1}^{M} \sum_{l_2=1}^{N} \cdots \sum_{l_M=1}^{N} \tilde{v}^2_{l_1\ldots l_M}(\boldsymbol{\omega}) + \tilde{v}^2_{l_1\ldots l_M}(-\boldsymbol{\omega}). \tag{69}$$

We define a set of normalized angle functions $v_{l_1\ldots l_M}(\boldsymbol{\omega})$ as follows

$$v_{l_1\ldots l_M}(\boldsymbol{\omega}) = \frac{\tilde{v}_{l_1\ldots l_M}(\boldsymbol{\omega})}{V(\boldsymbol{\omega})} \tag{70}$$

Because of the normalization, the sum of square of all $v_{l_1\ldots l_M}(\boldsymbol{\omega})$ is equal to 1.

A set of M-D window functions that are 2π periodic in M dimensional space is constructed from $w_0(\boldsymbol{\omega})$, $w_1(\boldsymbol{\omega})$ and $v_{l_1\ldots l_M}(\boldsymbol{\omega})$ as follows:

$$u_0(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^M} w_0(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) \tag{71}$$

$$u_{l_1\ldots l_M}(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^M} v_{l_1\ldots l_M}(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) w_1(\boldsymbol{\omega} + 2\boldsymbol{n}\pi). \tag{72}$$

Similar to the 2-D case (38), these functions also form a partition of unity in M-D space

$$u_0^2(\boldsymbol{\omega}) + \sum_{l_1=1}^{M} \sum_{l_2=1}^{N} \cdots \sum_{l_M=1}^{N} u^2_{l_1\ldots l_M}(\boldsymbol{\omega}) + u^2_{l_1\ldots l_M}(-\boldsymbol{\omega}) = 1. \tag{73}$$

Example 3, UDCT in Three Dimensions: For illustration, the 3-D UDCT is considered as a particular case of M-D UDCT. In 3-D case, the three-variable angle functions $\tilde{v}_{l_1 l_2 l_3}(\omega_1, \omega_2, \omega_3)$, $l_2, l_3 = 1, \ldots, N$, correspond to M-D functions in (68), are

$$\tilde{v}_{1 l_2 l_3}(\boldsymbol{\omega}) = v_{l_2}(\omega_1, \omega_2) v_{l_3}(\omega_1, \omega_3) \tag{74}$$

$$\tilde{v}_{2 l_2 l_3}(\boldsymbol{\omega}) = v_{l_2}(\omega_2, \omega_1) v_{l_3}(\omega_2, \omega_3) \tag{75}$$

$$\tilde{v}_{3 l_2 l_3}(\boldsymbol{\omega}) = v_{l_2}(\omega_3, \omega_1) v_{l_3}(\omega_3, \omega_2). \tag{76}$$

These functions cover the three infinite pyramids in Fig. 15(a), (b), (c). Following the normalization in (69) and (70), we define the set of $3 \times N^2$ $v_{l_1 l_2 l_3}(\boldsymbol{\omega})$ functions in 3-D space whose sum of square is equal to one:

$$v_{l_1 l_2 l_3}(\boldsymbol{\omega}) = \frac{\tilde{v}_{l_1 l_2 l_3}(\boldsymbol{\omega})}{V(\boldsymbol{\omega})}. \tag{77}$$

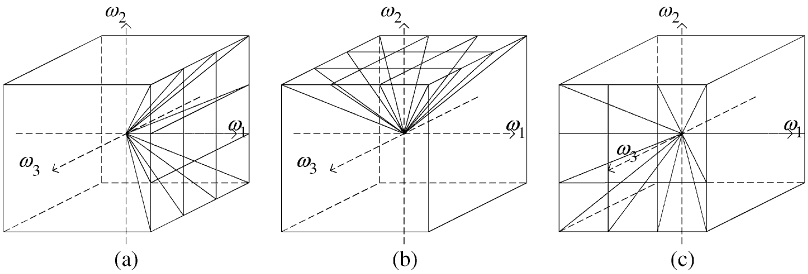


Fig. 15. Construct the angle functions in 3-D space. (a) 3-D angle functions in the first pyramid. (b) 3-D angle functions in the second pyramid. (c) 3-D angle functions in the third pyramid.

Similar to (71) and (72), a set of 3-D curvelet window functions is constructed from $w_0(\boldsymbol{\omega})$, $w_0(\boldsymbol{\omega})$, and $v_{l_1 l_2 l_3}(\boldsymbol{\omega})$ as follows:

$$u_0(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3} w_0(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) \tag{78}$$

$$u_{l_1 l_2 l_3}(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3} v_{l_1 l_2 l_3}(\boldsymbol{\omega} + 2\boldsymbol{n}\pi) w_1(\boldsymbol{\omega} + 2\boldsymbol{n}\pi). \tag{79}$$

Similar to the 2-D case in (38), these functions also form a partition of unity in 3-D space

$$u_0^2(\boldsymbol{\omega}) + \sum_{l_1=1}^{3}\sum_{l_2=1}^{N}\sum_{l_3=1}^{N} u_{l_1 l_2 l_3}^2(\boldsymbol{\omega}) + u_{l_1 l_2 l_3}^2(-\boldsymbol{\omega}) = 1. \tag{80}$$

B. Uniform Discrete Curvelet Transform in M Dimensions

From the set of M-D windows $u_0(\boldsymbol{\omega})$ and $u_{\boldsymbol{l}}(\boldsymbol{\omega})$, we define a M-D FB with analysis filters

$$F_0(\boldsymbol{\omega}) = 2^{\frac{M}{2}} u_0(\boldsymbol{\omega}) \tag{81}$$

$$F_{l_1 \ldots l_M}(\boldsymbol{\omega}) = 2^{\frac{(M-1)(n+1)}{2}+1} u_l(\boldsymbol{\omega}). \tag{82}$$

The decimation matrices for M dimensional signal are matrices of size $M \times M$. The decimation matrix for the lowpass branch $\boldsymbol{D}_0 = \operatorname{diag}\{2,\ldots,2\}$; The decimation matrices for the M groups of directional branches index by $l_1 l_2 \ldots l_M$ are $\boldsymbol{D}_{1 l_2 \ldots l_M} = \operatorname{diag}\{2, 2^{n+1},\ldots,2^{n+1}\}$, $\boldsymbol{D}_{2 l_2 \ldots l_M} = \operatorname{diag}\{2^{n+1}, 2, \ldots 2^{n+1}\}$, and continue until $\boldsymbol{D}_{M l_2 \ldots l_M} = \operatorname{diag}\{2^{n+1},\ldots,2^{n+1},2\}$. The synthesis side filters are the same as the corresponding analysis filters in all three dimensions.

Example 3, (continued). In 3-D case, the filters are defined from the curvelet windows in (81) and (82) as follows:

$$F_0(\boldsymbol{\omega}) = 2\sqrt{2} u_0(\boldsymbol{\omega}) \tag{83}$$

$$F_{l_1 l_2 l_3}(\boldsymbol{\omega}) = 2^{n+2} u_{l_1 l_2 l_3}(\boldsymbol{\omega}). \tag{84}$$

The M-D UDCT is constructed by cascading multiples of the M-D FB described above, with the condition that the number of directions at each level follows the rule of the curvelet transform. The derivation of the result M-D curvelet function is straightforward and is not included here. The index set for the curvelet function is similar to the 2-D case, except that the directional index $\boldsymbol{l}$ and position index $\boldsymbol{k}$ are now vectors with M components. The spatial M-D curvelet function has value oscillating along the direction defined by $\boldsymbol{l}$ index. The isosurfaces, on which the curvelet function has constant value, are approximately perpendicular to the direction $\boldsymbol{l}$. In order to show the 3-D wedge-shaped support in the frequency domain as well as the directionality in the spatial domain of 3-D curvelet functions, the isosurfaces of 9 curvelet functions are plotted in Fig. 16.

The structure of the M-D discrete uniform curvelet transform is similar to the one in Fig. 4. The M-D data are first transformed to the frequency domain. Then the data are multiplied with the set of directional windows defined above. The data are then wrapped into smaller M-D hypercube to reduce redundancy before being transformed back to the spatial domain.

We estimate the redundancy ratio for the UDCT in the general M dimensional case. Like other multiscale FB-based transform, the overcomplete ratio of the UDCT depends on number of scale J, but does not depend on the number of directions at each scale, or N_j. Let us consider the M-D data block being applied to the UDCT with 3 direction on each dimension (or $N_j = 3$). The FB has $M \times 3^{M-1}$ complex directional bands with decimation ratio 2^M. Therefore the number of real-valued coefficients is $M \times (3/2)^{M-1}$. The lowpass band also has a decimation ratio of 2^M. Assuming the transform is reiterated indefinitely, the redundancy is bounded by

$$\text{Red. UDCT MD} < M \times \left(\frac{3}{2}\right)^{M-1} \times \frac{1}{1-\left(\frac{1}{2}\right)^M} \tag{85}$$

or

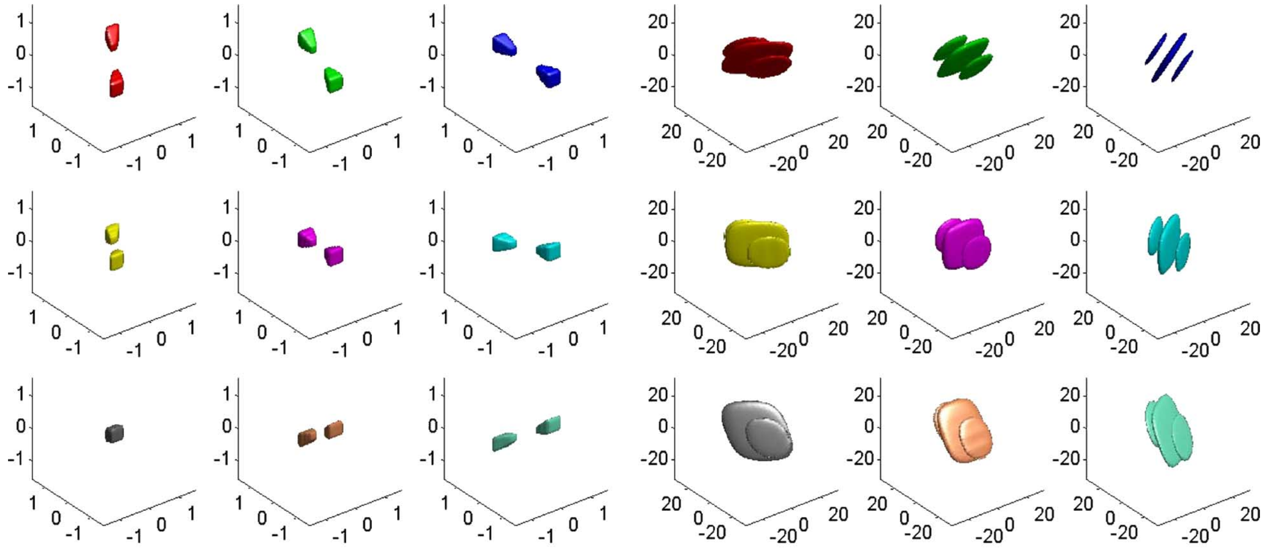
$$\text{Red. UDCT MD} < 2 \times M \times \frac{3^{M-1}}{2^M - 1} \tag{86}$$

Therefore, the 2-D UDCT has a redundancy ratio less than 4 and the 3-D UDCT has a redundancy ratio less than 54/7.

C. 3-D Curvelet Comparison

The 3-D FDCT is described in [30]. This implementation is a straightforward generalization of the FDCT. The 3-D curvelet windows are defined by multiplying two shearing angular windows with concentric square cubes. All window functions are then normalized to ensure that their sum is equal to one.

The contourlet transform also has a 3-D implementation in [31]. The transform is called the surfacelet transform. The structure of this transform is similar to the contourlet transform and is also implemented by a filter bank. The new feature in this generalization is a 3-D directional filter bank. Unlike the biorthogonal 2-D directional filter bank, this 3-D FB is 3 times overcomplete.



(a)

(b)

Fig. 16. Isosurface of 9 3-D uniform curvelet in (a) the frequency domain and (b) in the spatial domain.

TABLE I
NUMBER OF COEFFICIENTS AND REDUNDANCY RATIO FOR A DECOMPOSITION OF $64 \times 64 \times 64$ DATA

Name	3-D FDCT	Surfacelet	3-D UDCT
Configuration	[24 96]	[48 192]	[27 108]
Number of coeffs.	7.0e+6	8.9e+5	2.0e+6
Overcomplete ratio	≈ 27.0	≈ 3.4	≈ 7.6

TABLE II
DENOISING RESULTS ON LENA IMAGE, THE NUMBERS ON THE LEFT ARE REDUNDANT RATIO OF THE TRANSFORMS

Noise variance	10	20	30	40	50
Noisy PSNR	28.13	22.13	18.63	16.13	14.2
ContourletSD (2.33)	34.04	31.43	29.66	28.38	27.3
UDCT (4)	34.34	31.35	29.61	28.19	27.12
FDCT (7.6)	34.17	31.52	30.01	28.84	27.78
NSCT(53)	34.69	32.03	30.35	29.1	28.1

The three 3-D discrete curvelet transforms are used to decompose a synthetic seismic data cube of size $64 \times 64 \times 64$. All three decompositions have one low resolution and two directional resolutions. The overcomplete ratios of the three 3-D discrete transforms are displayed in Table I.

We believe that the overcomplete ratio is a key factor that distinguishes different curvelet implementations. The 3-D uniform curvelet has a redundancy twice higher compared to the surfacelet transform. Both of them have acceptable redundancy ratio. This is not the case for the current implementation of the FDCT.

VIII. IMPLEMENTATIONAL ASPECTS OF THE 2-D AND 3-D UDCT

A software package to implement the 2-D and 3-D UDCT in Matlab and Fortran has been developed at the Geoscience center, Mines ParisTech. Following are several practical aspects of the UDCT implementation.

A. *Implementation, Data Flow and Complexity of the UDCT*

In Section V, the UDCT is constructed as a multiresolution FB for the purpose of showing that the UDCT is indeed a forward and inverse discrete transform. Using FB theory, we estimate the curvelet basis functions and establish a clear forward and inverse transform as expressed in (64) and (63).

In practice, the UDCT implementation does not need to follow a multi-level structure as in Fig. 11(a); it is implemented directly like the structure in Fig. 11(b). All of the curvelet windows (or curvelet functions in frequency domain) are estimated at once based on the configuration (number of resolutions and number of directions for each resolution) of the transform, following (57) and (58). The set of $1 + \sum_{j=1}^{J} 2N_j$ curvelet windows (or filters in the frequency domain) of J resolutions are multiplied with the Fourier transform of data. The new set of windowed data is then wrapped to a rectangle according to the decimation rule in the frequency domain, as in (3). The decimation in the frequency domain is done by shifting and adding the windowed data based on the overall decimation ratio. For example, the windowed data in Fig. 17 is decimated by 4 in the row direction, and 8 in the column direction. The frequency data are partitioned into 32 smaller matrices, 4 in the row and 8 in the column directions. The non-zero matrices are added together. The curvelet coefficients for each resolution and each direction are recovered by inverse FFT applied on corresponding smaller frequency data.

The implementation of the UDCT by pre-computing all the curvelet windows has several advantages. Once all the windows are computed, the actual forward and inverse UDCT computations are straightforward. The complexity of the transform is equal to the redundancy ratio multiplied with the complexity of FFT on data, because the complexity of window multiplication and decimation, as in Fig. 4, is negligible.

Another important advantage of the UDCT is that the computational and storage of the set of $1 + \sum_{j=1}^{J} 2N_j$ curvelet

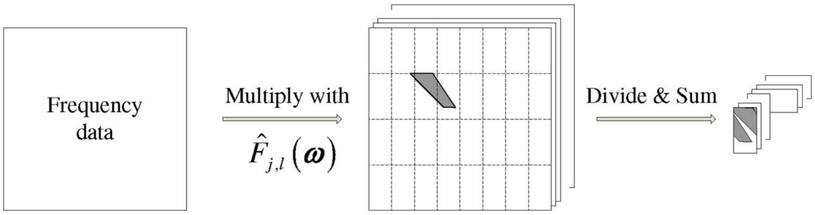


Fig. 17. Decimation in the frequency domain of the UDCT. At each scale there are $2N_j$ windows, the first and second N_j data matrices are divided by $\boldsymbol{D}_{j,0}$ in (59) and $\boldsymbol{D}_{j,1}$ (60), respectively.

windows is not a significant part of the transform. In fact, because of the inherent symmetry in each set of $2N_j$ windows in each scale, only $N_j/2$ windows need to be computed. The other $3N_j/2$ windows are obtained from the pre-computed windows by flipping and rotating operations. Moreover, all window functions are zero outside a small support region. By using sparse matrix format, the memory needed for storing $1+\sum_{j=1}^{J}2N_j$ windows is only about a fraction of the data storage. Efficient memory handling of the UDCT is very useful in processing 3-D transform, when the size of the data can be in the order of the system memory.

B. Data Size and Parallel Structure

An implicit assumption in all implementations of the curvelet transform is that the shape of input data is square (for 2-D) or cubic (for 3-D), and its dimensions are powers of 2. However, in industrial applications, the typical data can be of any size. The current implementation of UDCT accommodate data of different size by indexing the directional curvelet depending on its dimension. At each scale, the UDCT may have different numbers of directions on each dimension, which means there are M values of N_j, denoted by $N_j^{(m)}$. The index for the curvelet direction $\boldsymbol{l}$ has M elements, similar to index $\boldsymbol{l}$ in (68); The first element l_1, $1 \leq l_1 \leq M$ corresponds to the dimension (or the pyramid that the curvelet direction belongs to) and the rest are directions within that pyramid. For example, in the 2-D case, the directional curvelets at a certain scale will be divided into 'mostly vertical' and 'mostly horizontal' curvelets. A 2-D image that has twice the number of columns as rows, can be decomposed by a UDCT having twice the number of 'mostly vertical' curvelet functions compared to 'mostly horizontal' curvelet functions.

The data in an actual application of a discrete curvelet transform is possibly very large. It can even be larger than the size of the memory available on the computer. The solution is to divide the data into multiple blocks to process in sequence or in parallel. A method of dividing data using overlapping tapered windows is proposed in [32] to be used with the FDCT. The same approach is used in our implementation of the UDCT.

C. Border Extension

The UDCT and FDCT are based on the FFT transform, assuming that the image is periodically extended across its boundaries. Therefore, the discrete curvelet basis function is periodically extended around the opposite border of the finite size discrete 2-D data (curvelet 1 in Fig. 18). This may lead to potential border artifacts in practical applications. For example, a significant curvelet coefficient corresponding to a curvelet function located near the border of a 2-D signal may actually corresponds to an event near the opposite border.

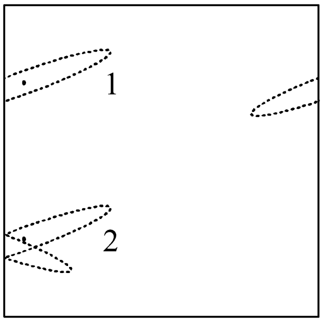
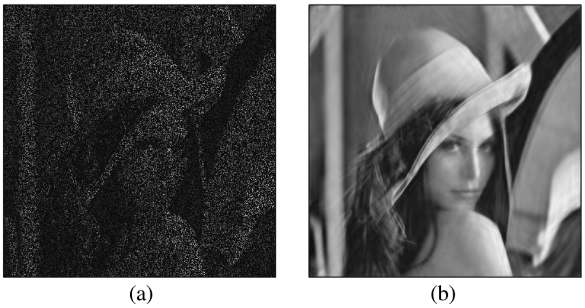


Fig. 18. Support of curvelet functions near borders of size-limited images: (1) is the case of periodic extension and (2) is the case of symmetric extension.

A proposed solution to this problem is to replace the FFT with the Discrete Cosine Transform (DCT) as in [33]. Because the DCT uses symmetric extension at the border of the transformed images, the curvelet functions in [33] are mirror-extended around the border of the finite size processed data (curvelet 2 in Fig. 18). However, this approach has not completely solved the potential border artifact problem. By definition, the curvelet function is highly anisotropic at high frequencies, with one dominant direction. Because of the use of the DCT, the curvelet basis near the border is also mirrored around the border edges. Thus they have two dominant directions and may not be considered as a 'true' curvelet. The potential artifact problem is that a significant curvelet coefficient belonging to a subband of a certain dominant direction may actually correspond to events at another direction that happens to mirror the direction of the subband.

A practical general solution is to taper the input data at the border. The processed data usually do not have the size optimal for the FFT. For the UDCT implementation, the data are first extended to suitable size by repeating the first and last rows (or columns). It is then multiplied with a smooth window similar to $w_0(\boldsymbol{\omega})$ in (19). The window function has a smooth transition area from 0 to 1. This procedure leads to a slightly higher redundancy ratio. But it also removes the border artifact problem by guaranteeing two aspects. First, there is no significant feature near the border of the processed data. Second, there is no abrupt change at the border of the actual data.



(a) (b)

Fig. 19. Image inpainting application of the UDCT. (a) Lena image with 80% number of pixels missing $\mathrm{PSNR} = 6.28\ \mathrm{dB}$. (b) inpainted Lena image by algorithm in [36] with UDCT dictionary, $\mathrm{PSNR} = 26.44\ \mathrm{dB}$.

IX. Numerical Results

In this section, we present several simple experiments using the UDCT. The parameters η_a and η_b receive values as in (48). The UDCT has also been used in other image processing applications [34], [35].

A. Application of the UDCT to Image Inpainting

Here we demonstrate the use of the UDCT in the image inpainting problem. We use the image inpainting algorithm in [36], publicly available with a Matlab implementation. The inpainting method is designed for filling missing pixels in images; The main assumption of the method is that the image consists of several layers and all layers have sparse representations in incoherent dictionaries. We assume that the testing image has a sparse representation in the curvelet domain. The original code employs the FDCT dictionary which is replaced by the UDCT dictionary. The number of randomly missing pixels in Lena image is about 80% of the total number of pixels in the image. The inpainted result is nearly identical as the one produced by using FDCT dictionary (see Fig. 19). The peak signal to noise ratio (PSNR) of the inpainted images by UDCT and FDCT dictionaries are 26.44 and 26.56 dB, respectively.

B. Image and Video Denoising by Simple Thresholding

1) Image Denoising: The wavelet-type transforms are widely used for image denoising. Denoising by simple thresholding also provides a simple and effective indication of the performance of a discrete transform. We compared the denoising performance of four directional image representations: the contourlet transform (new implementation in [26]), the UDCT, the FDCT and the nonsubsampled contourlet transform (NSCT). The Lena testing image is contaminated with Gaussian white noise. The noisy image is transformed by a five-level UDCT. The UDCT subband coefficients are hard-thresholded at 3 times the noise variances in the subband. For the other three transforms we use the same experiment set up as in [18]. The results show that the denoising performances are related to the redundancy ratio. The UDCT and the new implementation of the contourlet transform have approximately the same denoising performances; The FDCT provides slightly better result. Therefore, we position the UDCT as a transform that balances the advantages of the two decompositions: practicality and performance.

TABLE III
Denoising Result on Mobile Data Cube of Size $192 \times 192 \times 192$ as in [31]

Noise variance	30	40	50
Surfacelets [31] (3.4)	25.86	24.72	23.88
3-D UDCT (7.6)	26.63	25.1	23.93

2) Video Denoising: Using the 3-D curvelet transform to denoise video data should provide even better performance than in 2-D image denoising, because it exploits temporal correlation between different frames. We test the UDCT decomposition in denoising video data contaminated by white Gaussian noise.

The experiment setups are the same as in [31]. The video data is decomposed by the UDCT into one lowpass and four directional scales. The number of directional subbands from coarse to fine scales are 27, 27, 108, and 432. The hard thresholds are set at three times the noise variances, multiplied by the norm of the curvelet functions. As expected, the result in Table III shows that the performance of the UDCT is better than the surfacelet transform. The improvement is due to the higher redundancy ratio and better frequency characteristics of the UDCT curvelets, compared to the filters used in the surfacelet FB. We do not include the FDCT in the experiment because with the current implementation, the redundancy is too high and it cannot run on our workstation.

X. Conclusion

We have presented a novel discrete curvelet transform, which is called Uniform Discrete Curvelet Transform. It can be considered an ‘engineering’ approach to implement the curvelet transform. It has several advantages over existing transforms, such as lower redundancy ratio, hierarchical data structure and ease of implementation. Specifically the UDCT coefficients have a clear parent children structure and the UDCT functions have closed-form expressions in the discrete domain. Applications of the transform using these advantages are subjects of ongoing research in academic and industrial environments.

Acknowledgment

The authors would like to thank F. ten Kroode (Shell E&P), M. Noble, P. Podvin (Mines ParisTech), and R. Dyer (Fugro Seismic Imaging) for fruitful discussions.

References

[1] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. New York: Academic, 1999.

[2] M. Vetterli, “Wavelets, approximation and compression,” *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 59–73, 2001.

[3] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, “Shiftable multiscale transform,” *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.

[4] N. G. Kingsbury, “Complex wavelets for shift invariant analysis and filtering of signals,” *J. Appl. Computat. Harmon. Anal.*, vol. 10, no. 3, pp. 234–253, May 2001.

[5] N. G. Kingsbury, “Image processing with complex wavelets,” *Phil. Trans. Royal Soc. London A*, vol. 357, no. 1760, pp. 2543–2560, Sept. 1999.

[6] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying, “Fast discrete curvelet transforms,” *Multiscale Model. Simul.*, vol. 5, no. 3, pp. 861–899, 2006.

[7] G. Easley, D. Labate, and W.-Q. Lim, “Sparse directional image representations using the discrete shearlet transform,” *Appl. Computat. Harmon. Anal.* vol. 25, pp. 25–46, Jul. 2008 [Online]. Available: http://www.shearlet.org/

[8] M. N. Do and M. Vetterli, “The contourlet transform: An efficient directional multiresolution image representation,” *IEEE Trans. Image Process.*, vol. 14, pp. 2107–2116, Dec. 2005.

[9] E. J. Candès and D. L. Donoho, “Curvelets: A surprisingly effective nonadaptive representation for objects with edges,” in *Curve and Surface Fitting: Saint-Malo 99*, A. Cohen, J.-L. Merrien, and L. L. Schumaker, Eds. Nashville, TN: Vanderbilt Univ. Press, 2000, pp. 105–120.

[10] E. J. Candès and L. Demanet, “The curvelet representation of wave propagators is optimally sparse,” *Commun. Pure Appl. Math.s*, vol. 58, no. 11, pp. 1472–1528, 2005.

[11] H. Douma and M. V. de Hoop, “Leading-order seismic imaging using curvelets,” *Geophys.*, vol. 72, no. 6, pp. S231–S248, 2007.

[12] H. Chauris and T. T. Nguyen, “Seismic demigration/migration in the curvelet domain,” *Geophys.*, vol. 73, no. 2, pp. S35–S46, 2008.

[13] F. Herrmann, D. Wang, G. Hennenfent, and P. Moghaddam, “Curvelet-based seismic data processing: A multiscale and nonlinear approach,” *Geophys.*, vol. 73, no. 1, pp. A1–A5, 2008.

[14] J.-L. Starck, E. J. Candès, and D. L. Donoho, “Gray and color image contrast enhancement by the curvelet transform,” *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 706–717, Jun. 2003.

[15] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, “Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA),” *J. Appl. Computat. Harmon. Anal.*, vol. 19, pp. 340–358, Nov. 2005.

[16] J. Ma and G. Plonka, “Combined curvelet shrinkage and nonlinear anisotropic diffusion,” *IEEE Trans. Image Process.*, vol. 16, pp. 2198–2206, Sep. 2007.

[17] H. Chauris, “Seismic imaging in the curvelet domain and its implications for the curvelet design,” in *Proc. 73th SEG Ann. Meet. Expo., Expand. Abstr.*, 2006.

[18] A. L. Cunha, J. Zhou, and M. N. Do, “The nonsubsampled contourlet transform: Theory, design, and applications,” *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, 2006.

[19] T. T. Nguyen and S. Oraintara, “The shiftable complex directional pyramid, part 1: Theoretical aspects,” *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4651–4660, Oct. 2008.

[20] E. J. Candès and D. L. Donoho, “New tight frames of curvelets and optimal representations of objects with C^2 singularities,” *Commun. Pure Appl. Math.*, vol. 57, pp. 219–266, 2004.

[21] P. Vaidyanathan*, Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[22] J.-L. Starck, E. J. Candès, and D. L. Donoho, “The curvelet transform for image denoising,” *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.

[23] F. J. Herrmann, U. Boeniger, and D. J. Verschuur, “Nonlinear primary-multiple separation with directional curvelet frames,” *Geophys. J. Int.*, vol. 170, pp. 781–799, 2007.

[24] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[25] T. T. Nguyen and S. Oraintara, “On the aliasing effect of the contourlet filter banks,” in *Proc. 14th Eur. Signal Process. Conf. (EUSIPCO 2006)*, Florence, Sep. 2006.

[26] Y. Lu and M. N. Do, “A new contourlet transform with sharp frequency localization,” in *Proc. Int. Conf. on Image Process. (ICIP 06)*, Oct. 2006, pp. 1629–1632.

[27] J. Kovačević and A. Chebira, “Life beyond bases: The advent of frames (part i),” *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 86–104, Jul. 2007.

[28] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, “The dual-tree complex wavelet transform,” *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123–151, Nov. 2005.

[29] F. C. A. Fernandes, R. L. C. van Spaendonck, and C. S. Burrus, “A new framework for complex wavelet transforms,” *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1825–1837, Jul. 2003.

[30] L. Ying, L. Demanet, and E. J. Candès, “3D discrete curvelet transforms,” *Proc. SPIE Conf. Wavelet Appl. Signal Image Process. XI*, Jul. 2005.

[31] Y. Lu and M. N. Do, “Multidimensional directional filter banks and surfacelets,” *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 918–931, 2007.

[32] D. Thomson, G. Hennenfent, H. Modzelewski, and F. Herrmann, “A parallel windowed fast discrete curvelet transform applied to seismic processing,” in *Proc. 73th SEG Ann. Meet. Expo., Expand. Abstr.*, 2006.

[33] L. Demanet and L. Ying, “Curvelets and wave atoms for mirror-extended images,” in *Proc. SPIE Conf. Wavelet Appl. Signal Image Process. XII*, Aug. 2007.

[34] A. Vo and S. Oraintara, “A study of relative phase in complex wavelet domain: Property, statistics and applications in texture image retrieval and segmentation,” *Signal Process.: Image Commun.*, vol. 25, pp. 28–46, 2010.

[35] Y. Rakvongthai and S. Oraintara, “Statistics and dependency analysis of the uniform discrete curvelet coefficients and hidden markov tree modeling,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS’09)*, Taipei, May 2009, pp. 525–528.

[36] M. J. Fadili, J. L. Starck, and F. Murtagh, “Inpainting and zooming using sparse representations,” *Comput. J.*, vol. 52, no. 1, pp. 64–79, 2009.

Truong T. Nguyen received the B.S. and the M.S. degrees in electrical engineering from the Hanoi University of Technology, Hanoi, Vietnam, in 1995 and 2001, respectively, and the Ph.D. degree in electrical engineering from the University of Texas at Arlington in 2006.

From 1995 to 2002, he was with the Alcatel NSV, Hanoi. From June 2006 to September 2008, he was a Postdoctoral Fellow with the Geosciences Department, Mines Paristech, Fontainebleau, France. Currently, he is with the R&D Department, Fugro Seismic Imaging, Swanley, U.K. His current research interests are in the field of signal processing and seismic imaging.

Hervé Chauris received the Ph.D degree in geophysics from the Paris school of Mines.

He spent five years with the Seismic Imaging Research Group, Shell, and then joined the Geophysical Team at Mines Paristech, Fontainebleau, France, in 2005. His research interests include seismic wave propagation, curvelets, and inverse problems.