# Ripplet: A new transform for image processing

Jun Xu *, Lei Yang, Dapeng Wu

*Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA*

## ARTICLE INFO

## ABSTRACT

Efficient representation of images usually leads to improvements in storage efficiency, computational complexity and performance of image processing algorithms. Efficient representation of images can be achieved by transforms. However, conventional transforms such as Fourier transform and wavelet transform suffer from discontinuities such as edges in images. To address this problem, we propose a new transform called ripplet transform. The ripplet transform is a higher dimensional generalization of the curvelet transform, designed to represent images or two-dimensional signals at different scales and different directions. Specifically, the ripplet transform allows arbitrary support $c$ and degree $d$ while the curvelet transform is just a special case of the ripplet transform (Type I) with $c = 1$ and $d = 2$. Our experimental results demonstrate that the ripplet transform can provide efficient representation of edges in images. The ripplet transform holds great potential for image processing such as image restoration, image denoising and image compression.

© 2010 Elsevier Inc. All rights reserved.

## 1. Introduction

Efficient representation of images or signals is critical for image processing, computer vision, pattern recognition and image compression. Harmonic analysis [1] provides a methodology to represent signals efficiently. Specifically, harmonic analysis is intended to efficiently represent a signal by a weighted sum of basis functions; here the weights are called coefficients and the mapping from the input signal to the coefficients is called transform. In image processing, Fourier transform is usually used. However, Fourier transform can only provide an efficient representation for smooth images but not for images that contain edges. Edges or boundaries of objects cause discontinuities or singularities in image intensity. How to efficiently represent singularities in images poses a great challenge to harmonic analysis. It is well known that one-dimensional (1D) singularities in a function (which has finite duration or is periodic) destroy the sparsity of Fourier series representation of the function, which is known as Gibbs phenomenon. In contrast, wavelet transform is able to efficiently represent a function with 1D singularities [2,3]. However, typical wavelet transform is unable to resolve two-dimensional (2D) singularities along arbitrarily shaped curves since typical 2D wavelet transform is just a tensor product of two 1D wavelet transforms, which resolve 1D horizontal and vertical singularities, respectively.

To overcome the limitation of wavelet, ridgelet transform [4,5] was introduced. Ridgelet transform can resolve 1D singularities along an arbitrary direction (including horizontal and vertical direction). Ridgelet transform provides information about orientation of linear edges in images since it is based on Radon transform [6], which is capable of extracting lines of arbitrary orientation.

Since ridgelet transform is not able to resolve 2D singularities, Candes and Donoho proposed the first generation curvelet transform based on multi-scale ridgelet [7,8]. Later, they proposed the second generation curvelet transform [9,10]. Curvelet transform can resolve 2D singularities along smooth curves. Curvelet transform uses a parabolic scaling law to achieve anisotropic directionality. From the perspective of microlocal analysis, the anisotropic property of curvelet transform guarantees resolving 2D singularities along $C^2$ curves [11,9,10,12]. Similar to curvelet, contourlet [13,14] and bandlet [15] were proposed to resolve 2D singularities.

However, it is not clear why parabolic scaling was chosen for curvelet to achieve anisotropic directionality. Regarding this, we have two questions: is the parabolic scaling law optimal for all types of boundaries? If not, what scaling law will be optimal? To address these two questions, we intend to generalize the scaling law, which results in a new transform called *ripplet transform* Type I. Ripplet transform Type I generalizes curvelet transform by adding two parameters, i.e., support $c$ and degree $d$; hence, curvelet transform is just a special case of ripplet transform Type I with $c = 1$ and $d = 2$. The new parameters, i.e., support $c$ and degree $d$, provide ripplet transform with anisotropy capability of representing singularities along arbitrarily shaped curves. The ripplet transform has the following capabilities:

* Corresponding author. Fax: +1 352 392 0638.
  *E-mail address:* junxu9@ufl.edu (J. Xu).

- *Multi-resolution*: Ripplet transform provides a hierarchical representation of images. It can successively approximate images from coarse to fine resolutions.
- *Good localization*: Ripplet functions have compact support in frequency domain and decay very fast in spatial domain. So ripplet functions are well localized in both spatial and frequency domains.
- *High directionality*: Ripplet functions orient at various directions. With the increasing of resolution, ripplet functions can obtain more directions.
- *General scaling and support*: Ripplet functions can represent scaling with arbitrary degree and support.
- *Anisotropy*: The general scaling and support result in anisotropy of ripplet functions, which guarantees to capture singularities along various curves.
- *Fast coefficient decay*: The magnitudes of ripplet transform coefficients decay faster than those of other transforms', which means higher energy concentration ability.

Note that we have also developed ripplet transform Type II and Type III, which will be described in our future work.

To evaluate the performance of ripplet transform for image processing, we conduct experiments on synthetic and natural images in image compression and denoising applications. Our experimental results demonstrate that for some images, ripplet transform can represent images more efficiently than DCT and discrete wavelet transform (DWT), when the compression ratio is high. When used for image compression, ripplet transform based image coding outperforms JPEG for the whole bit rate range; and it achieves performance comparable to JPEG2000, when the compression ratio is high; but ripplet transform can provide better visual quality than JPEG2000. Our experimental results also show that the ripplet transform achieves superior performance in image denoising.

The remainder of the paper is organized as below. In Section 2, we review the continuous curvelet transform in spatial domain and frequency domain, and analyze the relations between them. In Section 3, we generalize the scaling law of curvelet to define ripplets and introduce continuous ripplet transform and inverse continuous ripplet transform. Then we discuss the discretization of ripplet transform in Section 4. We analyze ripplet functions from the perspective of frames in Section 5. Section 6 presents experimental results that demonstrate the good properties of ripplets. Section 7 concludes this paper and points out future research directions.

## 2. Continuous curvelet transform

Similar to the definition of wavelets, the whole curvelet family is constructed based on the element curvelet functions. The element curvelet functions vary from coarse to fine scales. The curvelet functions are translated and rotated versions of the element functions. The 2D curvelet function is defined as below [7,8]

$$\gamma_{a\vec{b}\theta}(\vec{x}) = \gamma_{a\vec{0}0}(R_\theta(\vec{x} - \vec{b})), \tag{1}$$

where $R_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ is the rotation matrix, which rotates $\theta$ radians. $\vec{x}$ and $\vec{b}$ are 2D vectors. $\gamma_{a\vec{0}0}$ is the element curvelet function.

The element curvelet function $\gamma_{a\vec{0}0}$ with scale parameter $a$ is defined in the frequency domain in polar coordinates [8].

$$\hat{\gamma}_a(r,\omega) = a^{3/4}W(a \cdot r)V(\omega/\sqrt{a}), \tag{2}$$

where $\hat{\gamma}_a(r,\omega)$ is the Fourier transform of $\gamma_{a\vec{0}0}$ in polar coordinate system. $W(r)$ is a 'radial window' and $V(\omega)$ is an 'angular window'. These two windows have compact supports on $[1/2,2]$ and $[-1,1]$, respectively. They satisfy the following admissibility conditions:

$$\int_{1/2}^{2} W^2(r)\frac{dr}{r} = 1, \tag{3}$$

$$\int_{-1}^{1} V^2(t)dt = 1. \tag{4}$$

These two windows partition the polar frequency domain into 'wedges' shown in Fig. 1.

From Eqs. (2)–(4), we know that the Fourier transform of curvelet function has a compact support in a small region which is the Cartesian product of $r \in [\frac{1}{2a}, \frac{2}{a}]$ and $\omega \in [-\sqrt{a}, \sqrt{a}]$. Curvelet also has small effective regions and decays rapidly in spatial domain. Compared to wavelets, in addition to the scaling information and position information, curvelet functions have another parameter to represent directional information. An intuitive way to obtain direction information is using rotated wavelet. However, the isotropic property of rotated wavelet transform makes the rotation unsuitable for resolving the wavefront set [9,10]. The parabolic scaling used in the definition of curvelet functions guarantees the effective length and width of the region to satisfy: *width* $\approx$ *length*$^2$ and leads to anisotropic behavior of curvelets, which makes curvelet transform suitable for resolving arbitrary wavefront. The parabolic scaling is the most important property of curvelet transform and also the key difference between the curvelet and the rotated wavelet.

Given a 2D integrable function $f(\vec{x})$, the continuous curvelet transform is defined as the inner product of $f(\vec{x})$ and the curvelet function [9,10,16]

$$C(a,\vec{b},\theta) = \langle f, \gamma_{a\vec{b}\theta}\rangle = \int f(\vec{x})\overline{\gamma_{a\vec{b}\theta}(\vec{x})}d\vec{x}, \tag{5}$$

where $C(a,\vec{b},\theta)$ are the curvelet coefficients and $(\bar{\cdot})$ denotes the conjugate operator. The curvelet coefficients describe the characteristics of signal at various scales, locations and directions.

In fact, the curvelet transform only captures the characteristics of high frequency components of $f(\vec{x})$, since the scale parameter $a$ cannot take the value of infinity. So the 'full' continuous curvelet transform consists of fine-scale curvelet transform and coarse-scale isotropic wavelet transform. The 'full' curvelet transform is invertible. We can perfectly reconstruct the input function based on its curvelet coefficients. With the 'full' curvelet transform, the Parseval formula holds [9,10,16]. If $f(\vec{x})$ is a high-pass function, it can be reconstructed from the coefficients obtained from Eq. (5) through

$$\tilde{f}(\vec{x}) = \int C(a,\vec{b},\theta)\gamma_{a\vec{b}\theta}(\vec{x})da\,d\vec{b}\,d\theta/a^3 \tag{6}$$
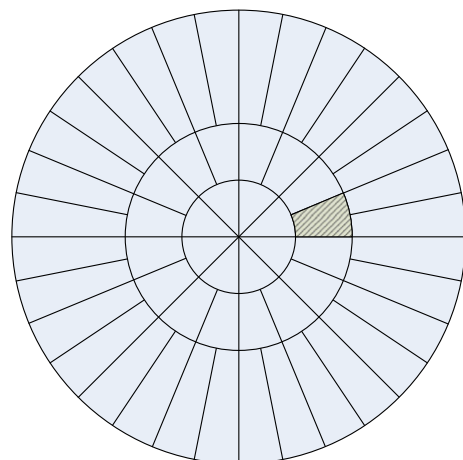


**Fig. 1.** The tiling of polar frequency domain. The shadowed 'wedge' corresponds to the frequency transform of the element function.

and

$$\|f\|^2 = \int |C(a, \vec{b}, \theta)|^2 da\, d\vec{b}\, d\theta/a^3. \tag{7}$$

## 3. Continuous ripplet transform

In this section, we introduce ripplet functions and continuous ripplet transform. We first generalize curvelet functions to define ripplet functions and then present the definition of continuous ripplet transform.

### 3.1. Ripplets

From the review in Section 2, we know that parabolic scaling used in curvelets leads to resolving of 2D singularities. However, there is no evidence to show that the parabolic scaling is the optimal scaling law. We can define the scaling law in a more broader scope and more flexible way. The ripplet function can be generated following the same strategy in Eq. (1)

$$\rho_{a\vec{b}\theta}(\vec{x}) = \rho_{a\vec{0}0}(R_\theta(\vec{x} - \vec{b})), \tag{8}$$

where $\rho_{a\vec{0}0}(\vec{x})$ is the ripplet element function and $R_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ is the rotation matrix. We define the element function of ripplet in frequency domain as

$$\hat{\rho}_a(r, \omega) = \frac{1}{\sqrt{c}} a^{\frac{1+d}{2d}} W(a \cdot r) V\left(\frac{a^{\frac{1}{d}}}{c \cdot a} \cdot \omega\right), \tag{9}$$

where $\hat{\rho}_a(r, \omega)$ are the Fourier transform of $\rho_{a\vec{0}0}(\vec{x})$. $W(r)$ is the 'radial window' on $[1/2, 2]$ and $V(\omega)$ is the 'angular window' on $[-1, 1]$. They also obey the admissibility conditions (3) and (4).

The set of functions $\{\rho_{a\vec{b}\theta}\}$ is defined as ripplet functions or ripplets for short, because in spatial domain these functions have ripple-like shapes. $c$ determines the support of ripplets and $d$ is defined as the degree of ripplets. Curvelet is just the special case of ripplet for $c = 1, d = 2$. Fig. 2 shows ripplets with different $c$ and different $d$ in spatial domain. From Fig. 2, we can see that ripplet functions decay very fast outside the effective region, which is an ellipse with the major axis pointing in the direction of the ripplet. The major axis is defined as the effective length and the minor axis, which is orthogonal to the major axis, is the effective width. The values of $c$ and $d$ will actually affect the effective length and width of ripplets in spatial domain. The effective region has the following properties for its length and width: $width \approx c \times length^d$. For fixed $d$, the larger $c$ is, the shorter the width is and the longer the length is. When $c$ is fixed and $d$ gets larger, the width gets shorter and the length is elongated. The customizable effective region tuned by support $c$ and degree $d$ bespeaks the most distinctive property of ripplets – the general scaling. For $c = 1, d = 1$, both axis directions are scaled in the same way. So ripplet with $d = 1$ will not have the anisotropic behavior. For $d > 1$, the anisotropic property is reserved for ripplet transform. For $d = 2$, ripplets have parabolic scaling; For $d = 3$, ripplets have cubic scaling; and so forth. Therefore, the anisotropy provides ripplets the capability of capturing singularities along arbitrary curves.

The ripplets as the generalization of curvelet have almost all the properties of curvelet except the parabolic scaling. Ripplets can get multi-resolution analysis of data. For each scale, ripplets have different compact supports such that ripplets can localize the singularities more accurately. Ripplets are also highly directional to capture the orientations of singularities.

### 3.2. Continuous ripplet transform

For a 2D integrable function $f(\vec{x})$, the continuous ripplet transform is defined as the inner product of $f(\vec{x})$ and ripplets

$$R(a, \vec{b}, \theta) = \langle f, \rho_{a\vec{b}\theta} \rangle = \int f(\vec{x}) \overline{\rho_{a\vec{b}\theta}(\vec{x})} d\vec{x}, \tag{10}$$

where $R(a, \vec{b}, \theta)$ are the ripplet coefficients. When the ripplet function intersects with curves in images, the corresponding coefficients will have large magnitude, and the coefficients decay rapidly along the direction of singularity as $a \to 0$.

The ripplet transform defined in Eq. (10) has the same issues as curvelet transform does, which is that the continuous ripplet transform can only capture the behavior of $f(\vec{x})$ in high frequency bands. To establish the 'full' continuous ripplet transform, we need to apply isotropic wavelet transform to represent the low frequency information. However, what really matters is the behavior of the transform in the high frequency bands, where the difference between curvelet and ripplet lies.

Now we transform images into another domain that we call ripplet domain. The challenges arise when we try to reconstruct images from ripplet coefficients. The theorems below introduce the inverse ripplet transform.

**Theorem 1.** Let $f \in L^2$ be a high-pass function, which means that its Fourier transform vanishes for $|\omega| < 2/a_0$ and $a_0$ is a constant. $f$ can be reproduced by its ripplet transform through

$$f(\vec{x}) = \int R(a, \vec{b}, \theta) \rho_{a\vec{b}\theta}(\vec{x}) da\, d\vec{b}\, d\theta/a^3, \tag{11}$$

And a Parseval formula for $f$ holds

$$\|f\|^2_{L_2} = \int |R(a, \vec{b}, \theta)|^2 da\, d\vec{b}\, d\theta/a^3. \tag{12}$$

**Proof.** For $r > 2/a_0$, (3) can be rewritten as

$$\int_{1/2}^2 W(a)^2 \frac{da}{a} = \int_0^{a_0 r} W(a)^2 \frac{da}{a} = \int_0^{a_0} W(a \cdot r)^2 \frac{da}{a} = 1. \tag{13}$$

Based on the admissibility condition, Eq. (3), we have

$$\int_0^{2\pi} V\left(\frac{a^{\frac{1}{d}}}{ca}(\omega - \theta)\right)^2 d\omega = ca^{\frac{d-1}{d}}. \tag{14}$$

For a special ripplet $\rho_{a\vec{0}\theta}(x)$, its Fourier transform has the property as below:

$$\int_0^{a_0} \int_0^{2\pi} |\hat{\rho}_{a00\theta}(\xi)|^2 d\theta \frac{da}{a^3} = \int_0^{a_0} \int_0^{2\pi} W(a \cdot r)^2 V\left(\frac{a^{\frac{1}{d}}}{ca}(\omega - \theta)\right)^2$$
$$\times \frac{1}{c} a^{\frac{1+d}{d}} d\theta \frac{da}{a^3} = 1. \tag{15}$$

Define

$$g_{a,\theta}(x) = \int \langle \rho_{a\vec{b}\theta}, f \rangle \rho_{a\vec{b}\theta} db. \tag{16}$$

We have $\rho_{a\vec{b}\theta} = \rho_{a\vec{0}\theta}(x - b)$, so

$$g_{a,\theta}(x) = \int \rho_{a\vec{0}\theta}(x - b) \left(\int \overline{\rho_{a\vec{0}\theta}(x - b)}(y - b) f(y) dy\right) db$$
$$= \int \rho_{a\vec{0}\theta}(x - b)(\overline{\rho_{a\vec{0}\theta}} * f)(b) db = (\rho_{a\vec{0}\theta} * \overline{\rho_{a\vec{0}\theta}} * f)(x). \tag{17}$$

According to the property of convolution, we can obtain the Fourier transform of $g$ as

$$\hat{g}_{a,\theta}(\omega) = \hat{\rho}_{a0\theta} \overline{\hat{\rho}_{a0\theta}} \hat{f}(\omega) = |\hat{\rho}_{a0\theta}(\omega)|^2 \hat{f}(\omega). \tag{18}$$

Using (15), we get

$$\int\int \hat{g}_{a,\theta}(\omega) d\theta \frac{da}{a^3} = \int\int |\hat{\rho}_{a0\theta}(\omega)|^2 \hat{f}(\omega) d\theta \frac{da}{a^3} = \hat{f}(\omega). \tag{19}$$
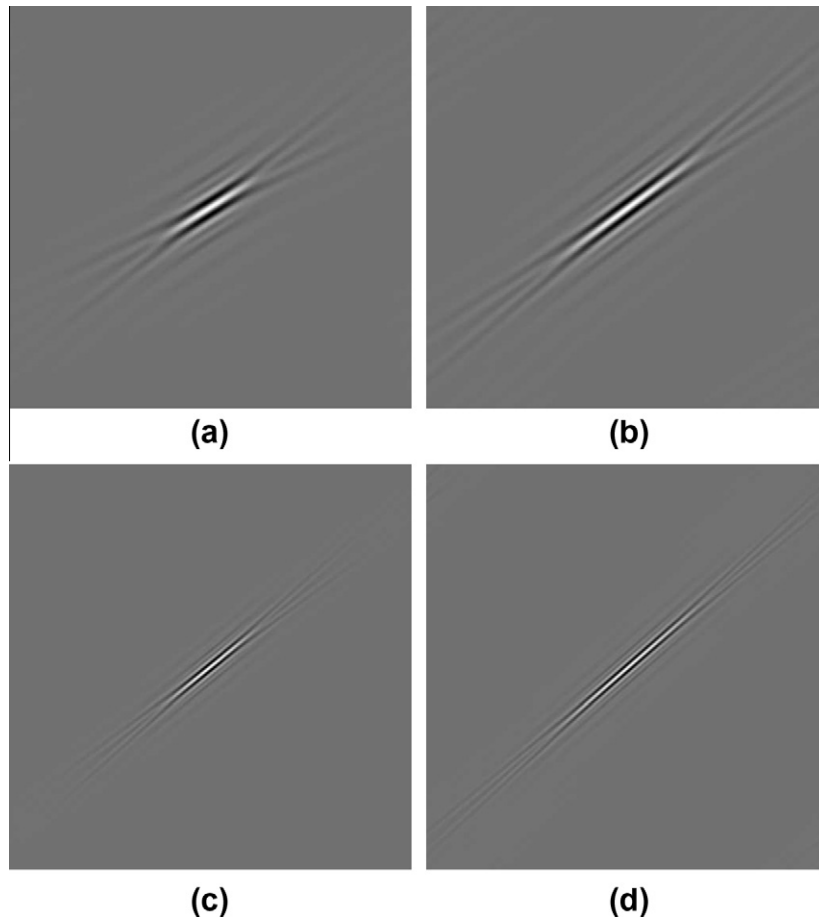
**Fig. 2.** Ripplets in spatial domain with different degrees and supports, which are all located in the center, i.e., $b = 0$. (a) $a = 3$, $\theta = 3\pi/16$, $c = 1$, $d = 2$, called curvelet particularly. (b) $a = 3$, $\theta = 3\pi/16$, $c = 1.5$, $d = 2$. (c) $a = 4$, $\theta = 3\pi/16$, $c = 1$, $d = 4$. (d) $a = 4$, $\theta = 3\pi/16$, $c = 1.5$, $d = 4$.

Further,

$$
f(x) = \int \hat{f}(\omega)e^{j\omega x}\, d\omega = \int\int\int \hat{g}_{a,\theta}(\omega)d\theta \frac{da}{a^3} e^{j\omega x}\, d\omega
$$

$$
= \int\int g_{a,\theta}(x)d\theta \frac{da}{a^3} = \int\int\int \langle \rho_{a\vec{b}\theta}, f\rangle \rho_{a\vec{b}\theta} \frac{da\,db\,d\theta}{a^3}. \tag{20}
$$

Using Plancherel formula and Eq. (15), we have

$$
\int |R(a,\vec{b},\theta)|^2 da\,d\vec{b}\,d\theta/a^3 = \int |(\overline{\rho_{a\vec{0}\theta}} * f)(b)|^2 da\,d\vec{b}\,d\theta/a^3
$$

$$
= \int |\hat{f}(\omega)|^2 |\hat{\rho}_{a0\theta}(\omega)|^2 d\theta \frac{da}{a^3} d\omega
$$

$$
= \int |\hat{f}(\omega)|^2 d\omega = \|f\|_{L_2}^2. \quad \square \tag{21}
$$

**Theorem 2.** *Let $f \in L^2$. There is a bandlimited purely radial function $\Phi$ in $L^2$ and of rapid decay so that, if $\Phi_{a_0,b}(x) = \Phi(x - b)$,*

$$
f(x) = \int \langle \Phi_{a_0,b}, f\rangle \Phi_{a_0,b}\, db + \int_0^{a_0}\int\int \langle f, \rho_{a\vec{b}\theta}\rangle \rho_{a\vec{b}\theta}(x)da\,d\vec{b}\,d\theta/a^3 \tag{22}
$$

*and*

$$
\|f\|^2 = \int |\langle \Phi_{a_0,b}, f\rangle|^2 db + \int_0^{a_0}\int\int |\langle f, \rho_{a\vec{b}\theta}\rangle|^2 da\,d\vec{b}\,d\theta/a^3. \tag{23}
$$

Since the issue of interest is just the fine-scale elements or high frequency bands, the choice of the wavelet transform for the coarse-scale can be very flexible. Similarly, Theorem 2 can be easily proved using the same arguments in [10].

## 4. Discrete ripplet transform

In the previous section, we introduced ripplets and continuous ripplet transform. Digital image processing needs discrete transforms instead of continuous transforms. Discretization of ripplet transform is proposed and analyzed in this section.

The discretization of continuous ripplet transform is actually based on the discretization of the parameters of ripplets, which is similar to discrete curvelet transform [16]. For the scale parameter $a$, we sample at dyadic intervals. The position parameter $b$ and rotation parameter $\theta$ are sampled at equal-spaced intervals. $a$, $\vec{b}$ and $\theta$ are substituted with discrete parameters $a_j$, $\vec{b}_k$ and $\theta_l$, which satisfy that $a_j = 2^{-j}$, $\vec{b}_k = [c \cdot 2^{-j} \cdot k_1,\ 2^{-j/d} \cdot k_2]^T$ and $\theta_l = \frac{2\pi}{c} \cdot 2^{-\lfloor j(1-1/d)\rfloor} \cdot l$, where $\vec{k} = [k_1,\ k_2]^T$, $(\cdot)^T$ denotes the transpose of a vector and $j, k_1, k_2,\ l \in \mathbb{Z}$. The degree of ripplets can take value from $\mathbb{R}$. Since any real number can be approximated by rational numbers, we can represent $d$ with $d = n/m$, $n, m \neq 0 \in \mathbb{Z}$. Usually, we prefer $n, m \in \mathbf{N}$ and $n, m$ are both primes. In the frequency domain, the corresponding frequency response of ripplet function is in the form

$$
\hat{\rho}_j(r,\omega) = \frac{1}{\sqrt{c}} a^{\frac{m+n}{2n}} W(2^{-j} \cdot r) V\left(\frac{1}{c} \cdot 2^{-\lfloor j\frac{m-n}{n}\rfloor} \cdot \omega - l\right), \tag{24}
$$

where $W$ and $V$ satisfy admissibility conditions as below:

$$
\sum_{j=0}^{+\infty} |W(2^{-j} \cdot r)|^2 = 1, \tag{25}
$$

$$
\sum_{l=-\infty}^{+\infty} \left| V\left(\frac{1}{c} \cdot 2^{-\lfloor j(1-1/d)\rfloor} \cdot \omega - l\right) \right|^2 = 1,\ \text{given } c,\ d \text{ and } j. \tag{26}
$$

The 'wedge' corresponding to the ripplet function in the frequency domain is

$$H_{j,l}(r, \theta) = \left\{ 2^j \leqslant |r| \leqslant 2^{2j}, \left| \theta - \frac{\pi}{c} \cdot 2^{-\lfloor j(1-1/d) \rfloor} \cdot l \right| \leqslant \frac{\pi}{2} 2^{-j} \right\}. \tag{27}$$

In discrete case, we can have better understanding about the parameters $c$ and $d$. The parameter $c$ controls the number of directions in the high-pass bands. $d$ controls how the number of directions changes across bands. For fixed $c$, $d$ helps to control the resolution in directions at each high-pass band. Given $d$, $c$ controls the number of directions at all high-pass bands. $c$ and $d$ determine the final number of directions at each band together.

The discrete ripplet transform of an $M \times N$ image $f(n_1, n_2)$ will be in the form of

$$R_{j,\vec{k},l} = \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{N-1} f(n_1, n_2) \overline{\rho_{j,\vec{k},l}(n_1, n_2)}, \tag{28}$$

where $R_{j,\vec{k},l}$ are the ripplet coefficients.

The image can be reconstructed through inverse discrete ripplet transform

$$\tilde{f}(n_1, n_2) = \sum_j \sum_{\vec{k}} \sum_l R_{j,\vec{k},l} \rho_{j,\vec{k},l}(n_1, n_2). \tag{29}$$

## 5. Tight frame

From the point of view of frame, ripplets provide a new tight frame with sparse representations for images with discontinuities along $C^d$ curves. Before we start the proof of tight frame, we introduce a lemma to facilitate the proof of the tight frame.

**Lemma 3.** *Suppose that* $\{\Phi\} \subset L^2(\mathbf{R}^2)$ *is a bandlimited function with*

$supp(\widehat{\Phi}) \subset [-\pi A, \pi A] \times [-\pi B, \pi B]$.

*Suppose that* $g \in L^2(\mathbf{R}^2)$ *is defined in the frequency domain by*

$\hat{g}(\omega) = |\widehat{\Phi}(\omega)|^2 \hat{f}(\omega)$

*where* $\hat{g}(\omega)$, $\widehat{\Phi}(\omega)$ *and* $\hat{f}(\omega)$ *are the Fourier transform of* $g$, $\Phi$ *and* $f$, *respectively.*

*If we have a set of functions* $\{\Phi_{\mathbf{k}}(\mathbf{x}) = \Phi(x_1 - k_1/A, x_2 - k_2/B)\}$, *then we have*

$$g(\mathbf{x}) = \sum_{\mathbf{k}} \langle f, \Phi_{\mathbf{k}} \rangle \Phi_{\mathbf{k}}(\mathbf{x})$$

*and*

$$\|g\|^2 = \sum_{\mathbf{k}} |\langle f, \Phi_{\mathbf{k}} \rangle|^2$$

**Proof.** Based on definition, we have $\Phi_{\mathbf{k}} = \Phi(\mathbf{x} - \mathbf{k})$. Since the Fourier transform of $g$ is the multiplication of $\widehat{\Phi}$ and $\hat{f}$, we have

$$g(\mathbf{x}) = \Phi * \overline{\Phi} * f(\mathbf{x}) = \sum_{\mathbf{k}} \Phi(\mathbf{x} - \mathbf{k})(\overline{\Phi} * f)(\mathbf{k})$$

$$= \sum_{\mathbf{k}} \Phi_{\mathbf{k}}(\mathbf{x}) \int \overline{\Phi(\mathbf{x} - \mathbf{k})} f(\mathbf{x}) d\mathbf{x} = \sum_{\mathbf{k}} \langle f, \Phi_{\mathbf{k}} \rangle \Phi_{\mathbf{k}}(\mathbf{x}), \tag{30}$$
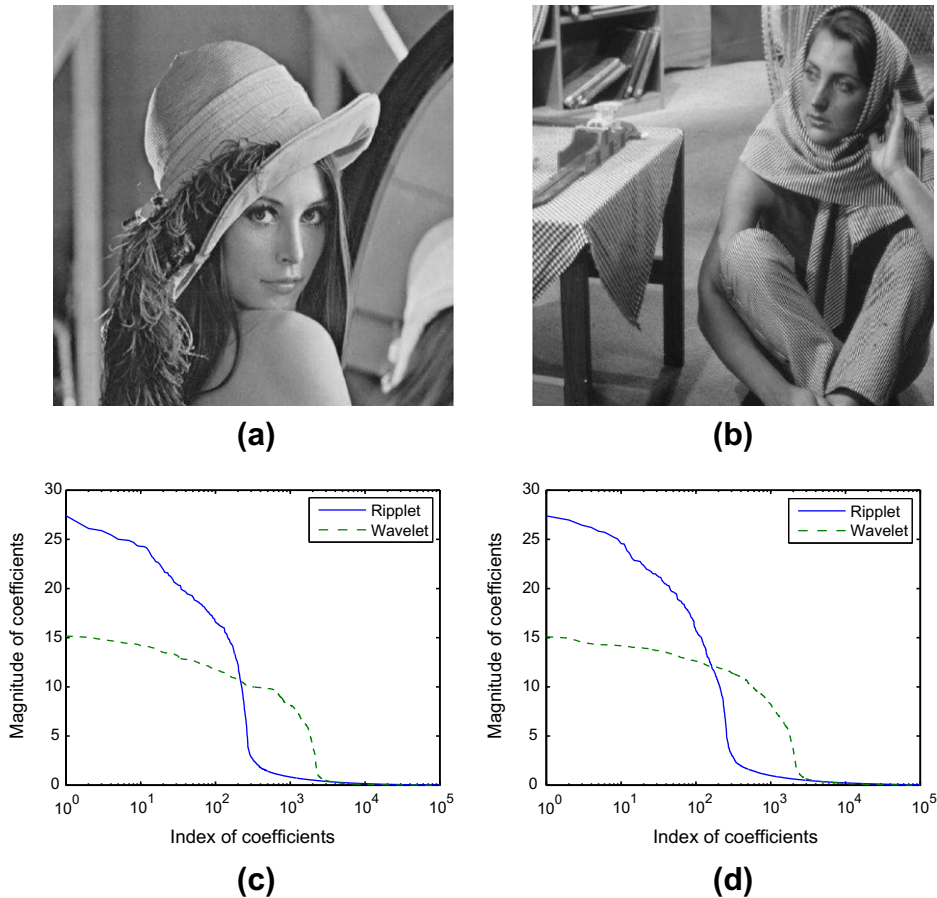


**Fig. 3.** The comparison of coefficients between ripplet transform and wavelet transform. (a) Lena, (b) Barbara, (c) coefficient decaying of Lena and (d) coefficient decaying of Barbara.

$$\|g\|_2^2 = \int g(\mathbf{x})\overline{g(\mathbf{x})}d\mathbf{x} = \int \sum_{\mathbf{k}}\langle f, \Phi_{\mathbf{k}}\rangle\Phi_{\mathbf{k}}(\mathbf{x})\sum_{\mathbf{k}}\overline{\langle f, \Phi_{\mathbf{k}}\rangle\Phi_{\mathbf{k}}(\mathbf{x})}d\mathbf{x}$$

$$= \sum_{\mathbf{k}}\sum_{\mathbf{k}}|\langle f, \Phi_{\mathbf{k}}\rangle|^2\int \Phi_{\mathbf{k}}(\mathbf{x})\overline{\Phi_{\mathbf{k}}(\mathbf{x})}d\mathbf{x}$$

$$= \sum_{\mathbf{k}}|\langle f, \Phi_{\mathbf{k}}\rangle|^2\sum_{\mathbf{k}}\int |\widehat{\Phi}_{\mathbf{k}}(\omega)|^2 d\omega. \tag{31}$$

From the definition of ripplets, all translated version of element ripplet will cover all the bands. Then we have $\sum_{\mathbf{k}}\int|\widehat{\Phi}_{\mathbf{k}}(\omega)|^2 d\omega = 1$. So

$$\|g\|_2^2 = \sum_{\mathbf{k}}|\langle f, \Phi_{\mathbf{k}}\rangle|^2. \quad \square \tag{32}$$

**Theorem 4.** *Ripplet functions provide a tight frame given any $L^2$ function f.*

$$\|f\|_{L_2}^2 = \sum_{j,\vec{k},l}|R(j,\vec{k},l)|^2. \tag{33}$$

The theorem can be proved with the translation parameter $\vec{b}$ and $l = 0$, based on the lemma above. For arbitrary $l$, we can rotate the coordinate to get a $\tilde{l} = 0$, where Lemma 3 applies.

## 6. Experimental results

In this section we present experimental results that demonstrate properties of ripplet transform and its potential applications.

### 6.1. Nonlinear approximation

To quantify the performance of sparse representation of transforms, nonlinear approximation (NLA) [1] of images is adopted as a common comparison approach. Suppose we have orthonormal basis $\{\phi_k\}$ and the corresponding coefficients $c_k = \langle g, \phi_k\rangle$. These coefficients are sorted in descending order with respect to the magnitude. The index $k$ is defined by

$$|c_0| \geqslant |c_1| \geqslant |c_2| \geqslant \cdots \geqslant |c_k| \geqslant \cdots \geqslant |c_{n-1}| \geqslant |c_n| \geqslant \cdots$$

The nonlinear approximation is obtained using $n$-largest coefficients as below:

$$g \approx \hat{g} = \sum_{i=0}^{n-1} c_i\phi_i. \tag{34}$$

Since ripplet transform provides a tight frame, the concentration of ripplet coefficients will lead to more accurate approximation in NLA. The faster the coefficients decay, the more compact energy will be allocated to the fewer large coefficients. To demonstrate the decay rate of ripplet transform coefficients, we first sort the ripplet coefficients with respect to their magnitudes and compare them to sorted wavelet coefficients in Fig. 3. It suggests that the coefficients of ripplet transform decay faster than those of wavelet transform. Similar patterns are observed for most standard test images.

We use peak signal-to-noise ratio (PSNR) versus number of retained coefficients to measure the quality of reconstructed images. PSNR is defined as
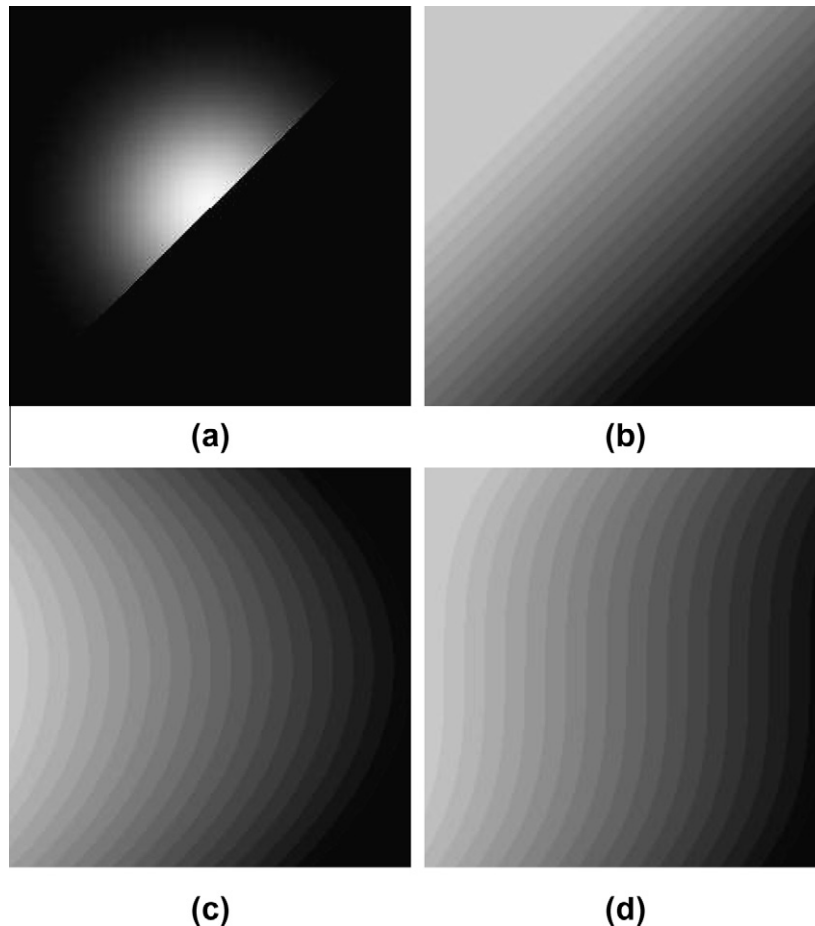


**Fig. 4.** Test images. (a) Truncated Gaussian image. (b) Multiple lines. (c) Parabolic curves. (d) Cubic curves.

$$PSNR = 10 \times \log_{10}\left(\frac{f_{max}^2}{mse}\right), \qquad (35)$$

where $f_{max}$ is the maximum value of image intensities and $mse$ is the mean square error between the reconstructed image $\tilde{f}_{M \times N}$ and original one $f_{M \times N}$:
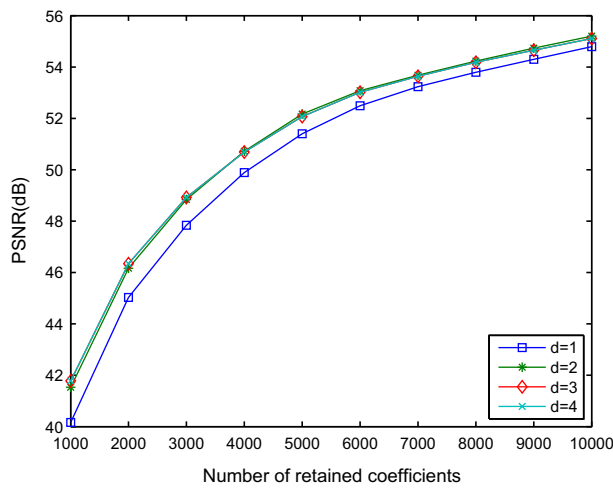
$$mse = \frac{1}{MN} \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{N-1} |f(n_1,n_2) - \tilde{f}(n_1,n_2)|^2. \qquad (36)$$

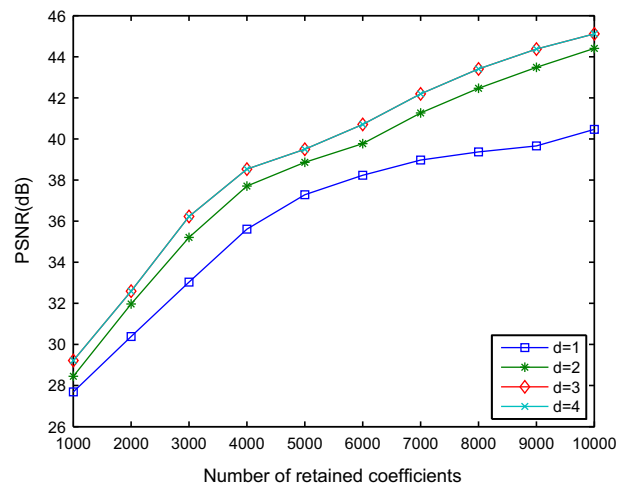### 6.1.1. Ripplets with different degrees

The images we used in the experiments are synthetic images with different edges which exhibit various 2D singularities along different curves shown in Fig. 4. Multiple lines and curves are synthesized with different coordinates to provide singularities along different curves. The truncated Gaussian image (Fig. 4(a)) presents a smooth changing part as well as singularity introduced by truncating. All images are of size $256 \times 256$ pixels.

The performance comparison between ripplets with fixed support ($c = 1$) but different degrees ($d = 1,2,3,4$) is shown in Fig. 5. To achieve the same PSNR, high degree ripplet needs fewer coefficients 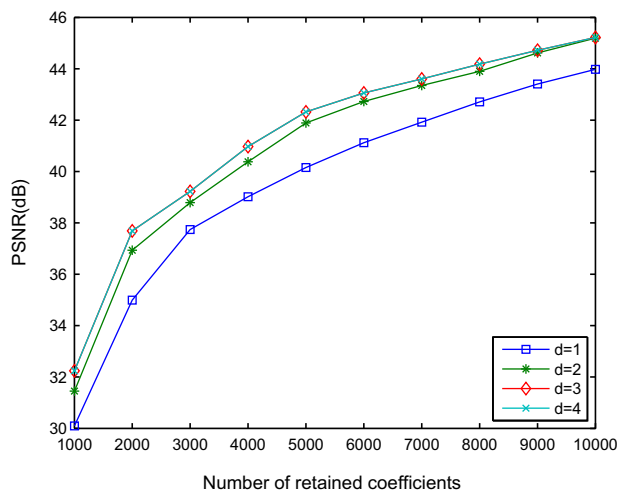than low degree ripplet. There is a big performance gap between degree 1 ripplet and others. For the same number of coefficients, ripplet with degree 1 achieves almost 2 dB lower than others in PSNR. In other words, the degree 1 ripplet needs more coefficients to achieve the same PSNR as other high degree ripplets. Degree 1 ripplet has isotropic behavior and is not directionally sensitive, whereas the other ripplets are anisotropic and can capture the singularities along curves in the test images. The gap between performance curves shows that the anisotropy helps a lot in representing 2D singularities efficiently. Fig. 5 also shows that ripplets with $c = 1, d > 2$ outperform curvelet (ripplet with $c = 1, d = 2$). Ripplets with $d = 4$ and $d = 3$ achieve the same highest PSNR for the same number of coefficients. Ripplet transform with $d \geqslant 3$ has more compact support and more directional sensitivity than curvelet, which can capture more accurate information about singularities. In our experiments, when $d > 3$, the performance is the same with $d = 3$. Since the discrete implementation of ripplet is based on the power of 2, the difference in performance brought by degree $d$ only appears in fine scales. The higher the degree is, the finer the scale is. Usually, for normal image size such as $256 \times 256$, $d = 3$ is the highest degree used in our experiments.
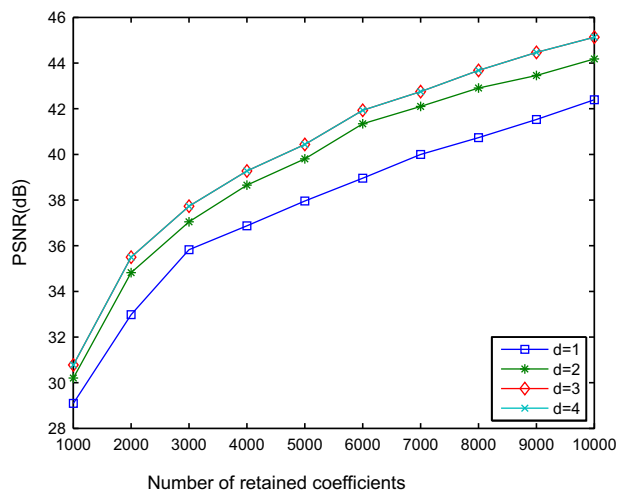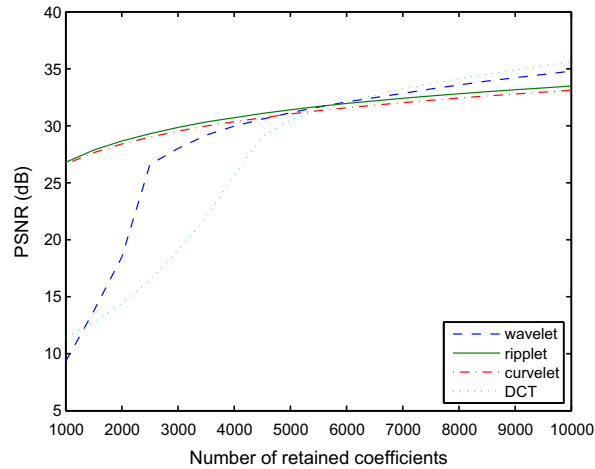


**Fig. 5.** Comparing nonlinear approximation performance of ripplets with fixed support ($c = 1$) and different degrees corresponding to the test images in Fig. 4. The $d = 2$ case is curvelet. (a) Truncated Gaussian function image. (b) Multiple lines. (c) Parabolic curves. (d) Cubic curves.

634

J. Xu et al. / J. Vis. Commun. Image R. 21 (2010) 627–639

### 6.1.2. Comparison with other transforms

To make comparison among different transforms, we present the results of nonlinear approximation using discrete wavelet transform, discrete cosine transform, discrete ripplet transform and discrete curvelet transform. The wavelet used in DWT is '9–7' biorthogonal wavelet [17,18]. DCT has the support of $8 \times 8$.

When sorting DCT coefficients, DCT coefficients from all blocks are considered together. The discrete ripplet transform uses ripplet with $c = 1$ and $d = 3$.

The results in Fig. 6(a) show that ripplet outperforms curvelet and can achieve the highest PSNR when the number of retained coefficients is less than 5500. Meanwhile, the reconstructed image



(a)



(b)



(c)



(d)



(e)



(f)

**Fig. 6.** (a) Performance comparison of nonlinear approximation using different transforms: DCT, DWT and discrete ripplet transform. (b) Original image, $512 \times 512$. (c) Ripplet ($c = 1$, $d = 3$) based NLA with 5000 largest coefficients, PSNR = 31.13 dB. (d) Curvelet (ripplet $c = 1$, $d = 2$) based NLA with 5000 largest coefficients, PSNR = 30.66 dB. (e) Wavelet based NLA with 5000 largest coefficients, PSNR = 30.13 dB. (f) DCT based NLA with 5000 largest coefficients, PSNR = 29.90 dB.
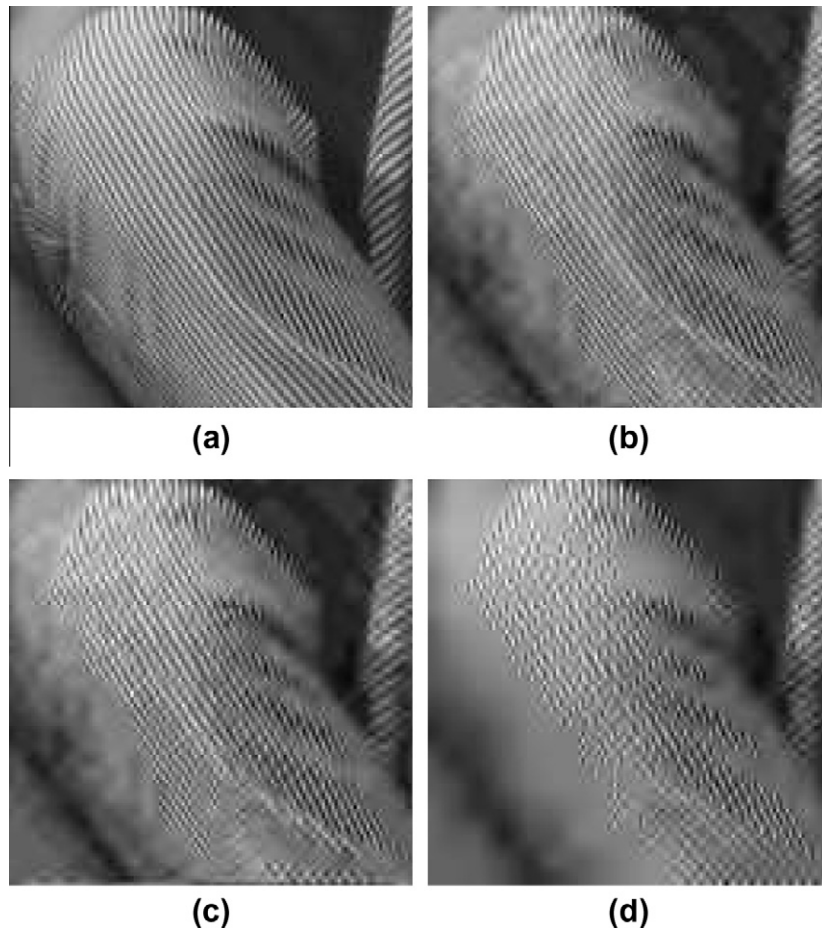
*J. Xu et al. / J. Vis. Commun. Image R. 21 (2010) 627–639*

635



**Fig. 7.** The visual quality comparison between ripplet based image codec and JPEG2000 for a patch cropped from 'Barbara', when bpp is equal to 0.3. (a) Original crop. (b) Ripplet $c = 1$, $d = 3$, PSNR = 25.39 dB. (c) Curvelet (ripplet $c = 1$, $d = 2$), PSNR = 24.12 dB. (d) JPEG2000, PSNR = 22.37 dB.
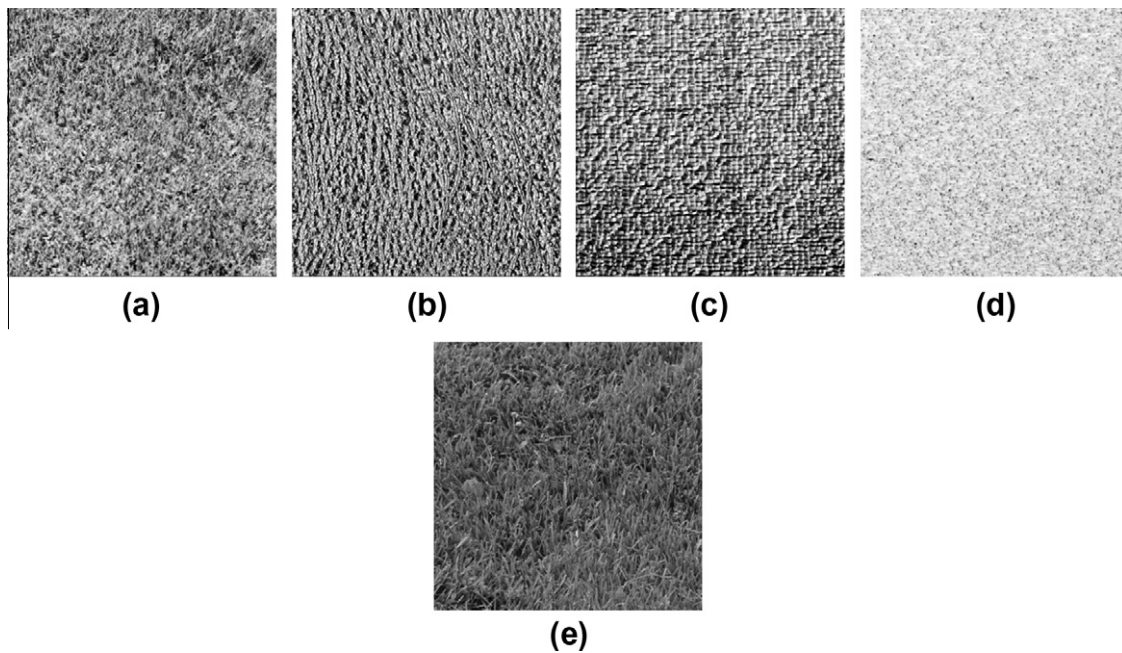


**Fig. 8.** Texture-rich images used in our experiment.

by ripplet transform as shown in Fig. 6(c) can provide better visual quality than DWT and DCT. We can see that ripplet avoids the 'ringing' artifacts of wavelet as shown in Fig. 6(e) and blocky artifacts of DCT as shown in Fig. 6(f). However, when using more coefficients, ripplet will no longer be the best. Therefore, ripplets have the strong capability of representing the structure of images with fewer number of coefficients.

## 6.2. Image compression

Inspired by the sparse representation of ripplet transform for images, we applied ripplet transform to image compression. The application of curvelet transform in image compression has been reported in [19], which demonstrated the advantage of curvelet over wavelet based image compression algorithm in terms of compress ratio vs PSNR. However, curvelet transform is an overcomplete transform and so is ripplet transform. It's more convincing to show the performance in real bitrate vs PSNR. In this experiment, we provide ripplet based image compression with real bitstreams and compare it to wavelet based approaches. For image compression application, we simply replaced the transform in a typical image coding scheme. The image compression codec we implemented consists of ripplet transform, quantization of ripplet coefficients, coefficient coding and entropy coding. In this implementation, uniform scalar quantizer was adopted. We employed the EBCOT [20] used in JPEG2000 [21] to code the coefficients, in which an adaptive binary arithmetic coder is used for entropy coding [22].

We compared the performance of ripplet, JPEG and JPEG 2000, for three cases, namely, the cropped image from 'barbara', the texture-rich images in the USC database [23] and natural images.

### 6.2.1. Comparison on the cropped image

In this experiment, we conduct comparisons on a cropped patch of size $128 \times 128$ pixels with rich texture from test image 'barbara'. The result shows that the ripplet based codec achieves 3 dB higher PSNR than JPEG2000 for the same bit-rate. Ripplet with $c = 1, d = 3$ can achieve higher PSNR than curvelet (ripplet with $c = 1, d = 2$) for the same bit-rate. In Fig. 7, we compare the subjective quality of the ripplet ($c = 1, d = 2, 3$) based codec and JPEG2000. It is obvious that the ripplet based codec preserves more details of texture, compared to JPEG2000.

### 6.2.2. Comparison on texture-rich images

We also tested the ripplet transform on texture-rich images of size $512 \times 512$ pixels given in Fig. 8. The ripplet transform searches over $(c, d)$ pairs for the highest PSNR. Results are listed in Table 1. As shown in Table 1, both ripplet and curvelet based codec achieves a slightly higher PSNR at low bit rate, compared to JPEG2000. Table 1 shows that ripplet outperforms curvelet, since curvelet is just a special case of ripplet.

### 6.2.3. Comparison on natural images

We compared the performance of ripplet transform on natural images of size $512 \times 512$ pixels. In this simulation, we used ripplet with $c = 1, d = 3$ and $c = 1, d = 2$ (e.g., curvelet). The results shown in

Fig. 9 indicate that the ripplet based codec outperforms JPEG by up to 3.3 dB on average at the same bit-rate. The ripplet with degree 3 outperforms curvelet (degree 2) as shown in Fig. 9(a) and (b). In Fig. 9(c), ripplet with degree 3 can achieve similar PSNR as curvelet
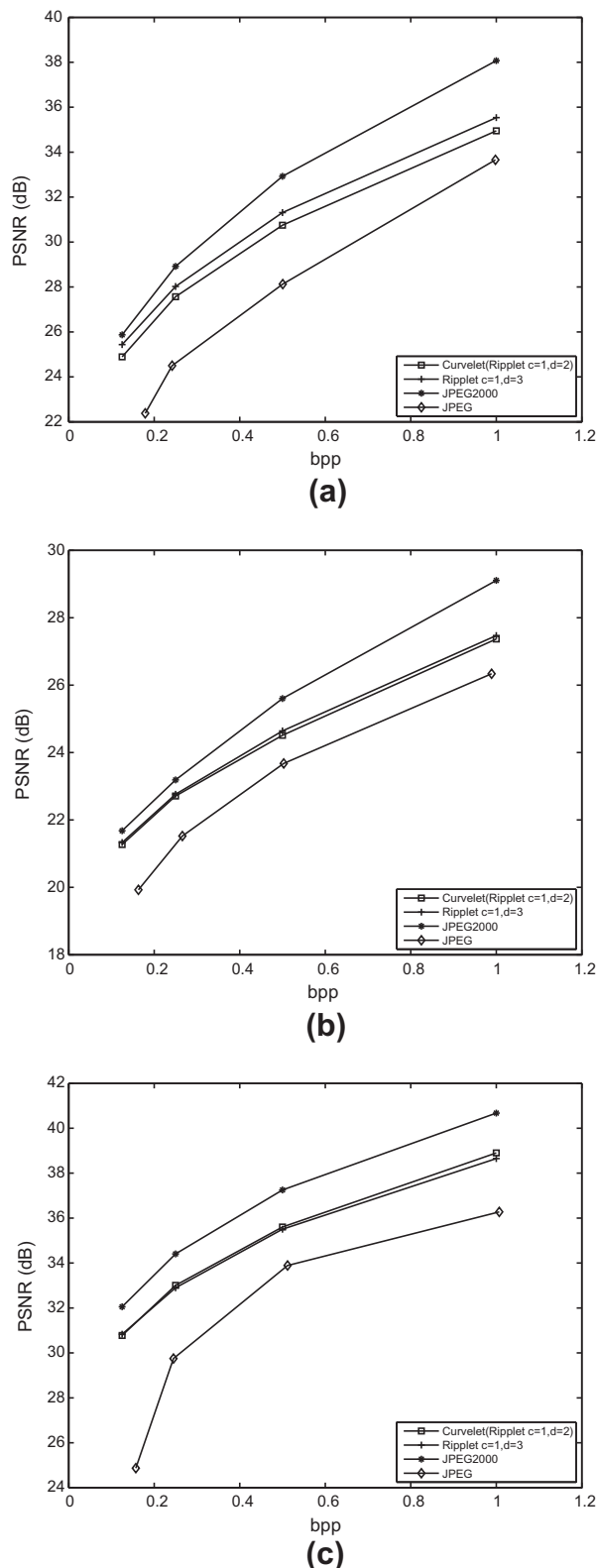


**(a)**



**(b)**



**(c)**

**Fig. 9.** PSNR vs. bpp for ripplet based image codec, curvelet based image codec, JPEG and JPEG2000. (a) Barbara. (b) Mandrill. (c) Tiffany.

**Table 1**
PSNR (dB) comparison of ripplet, curvelet and JPEG2000 at 0.03125 bpp.

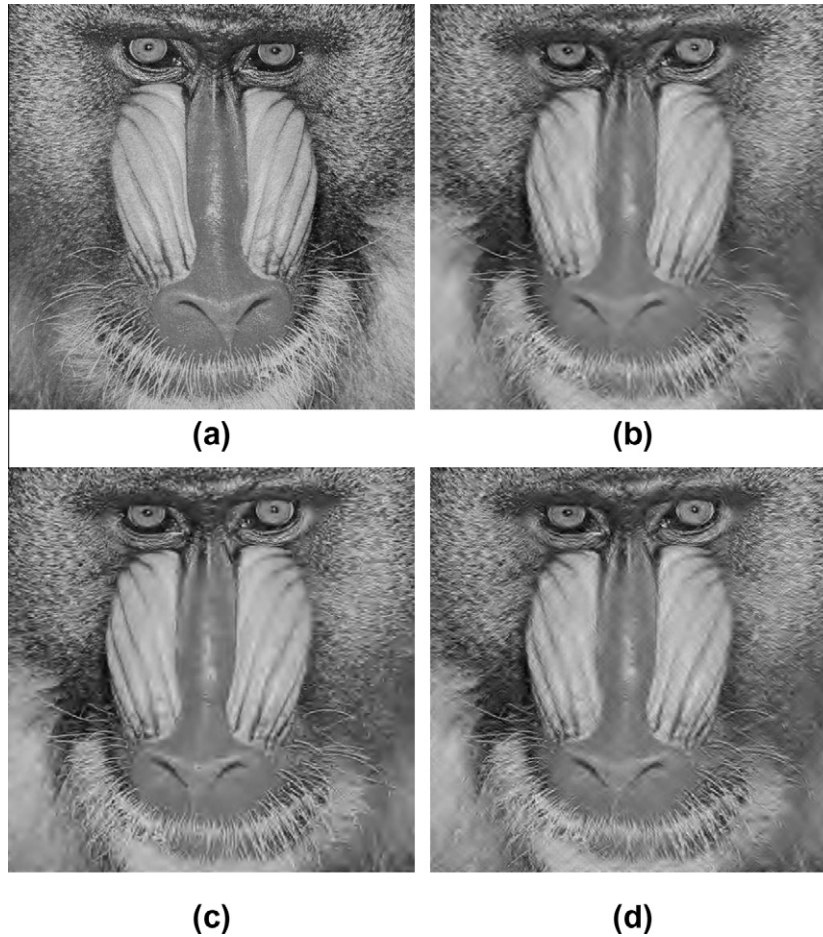| Texture images | Ripplet | JPEG 2000 | Curvelet |
|---|---|---|---|
| (a) | 14.87 | 14.76 | 14.87 |
| (b) | 11.58 | 11.45 | 11.54 |
| (c) | 11.77 | 11.59 | 11.77 |
| (d) | 20.82 | 20.72 | 20.81 |
| (e) | 21.13 | 20.96 | 21.13 |

**Fig. 10.** The visual quality comparison between ripplet based image codec, curvelet based image codec and JPEG2000 for 'mandrill', when bpp is equal to 0.25. (a) Original image. (b) Ripplet, PSNR = 22.76 dB. (c) JPEG2000, PSNR = 23.18 dB. (d) Curvelet, PSNR = 22.7 dB.

**Table 2**
Average PSNR gain of ripplet based codec, compared to JPEG, JPEG2000 and curvelet.

|  | Barbara | Mandrill | Tiffany |
|---|---|---|---|
| Average PSNR gain (dB) over JPEG | 2.9 | 1.2 | 3.3 |
| Average PSNR gain (dB) over JPEG2000 | −1.3 | −0.8 | −1.5 |
| Average PSNR gain (dB) over curvelet | 0.54 | 0.09 | 0 |

does, especially in low bit-rate. Compared to JPEG2000, the ripplet based codec achieves about 1 dB lower PSNR on average at the same bit-rate. However, the ripplet based codec can provide better subjective quality as shown in Fig. 10. When compression ratio is high, there are a lot of white spots around the face in the image coded by JPEG2000 in Fig. 10(b), while no obvious artifacts appear in the image coded using ripplet transform in Fig. 10(a). Moreover, the ripplet based codec keeps more details around the beard in 'mandrill' than JPEG2000 does. In this experiment, we presented results of ripplet with $c = 1, d = 3$ and $c = 1, d = 2$. To achieve the best performance, a set of pairs of $(c, d)$ can be tested and we can pick up the pair yielding the highest PSNR for given bitrate. Table 2 shows comparisons between the best performance of ripplet and performance of curvelet, JPEG and JPEG2000.

### 6.3. Image denoising

The proposed ripplet transform can be applied as a new method for noise removal in signals and images. Suppose an image $f(n_1, n_2)$ is corrupted by the additive noise,

$$g(n_1, n_2) = f(n_1, n_2) + n(n_1, n_2), \tag{37}$$

where $n(n_1, n_2)$ are independent, identically distributed Gaussian random variables with zero mean and variance $\sigma^2$.

Image denoising algorithms vary from simple thresholding to complicate model based methods. Since ripplet transform provides an overcomplete but sparse representation of images, simple hard thresholding in ripplet transform domain can remove most of noise. In our experiments, we use the following hard thresholding scheme: in the transform domain, a coefficient whose magnitude is smaller than the pre-determined threshold is set to zero; otherwise, the coefficient is unchanged. Then we reconstruct the image by inverse transform. In the experiments of this paper, we search over a selected range for the optimal threshold that provides the highest PSNR. To ensure a fair comparison, we apply the same optimal thresholding searching strategy to other transforms to be compared with.

As shown in Fig. 11, ripplet transform can achieve higher PSNR than curvelet and DWT. Meanwhile, the ripplet transform (Fig. 11(c)) can restore the edges better than curvelet (Fig. 11(d)) and DWT (Fig. 11(e)). The reason is that ripplet transform can represent these edges very sparsely, whereas noise will have small values in all ripplet coefficients. Then hard thresholding can remove the noise with little damage to images. On the other hand, wavelet transform cannot represent edges well; so edges are blurred due to hard thresholding.
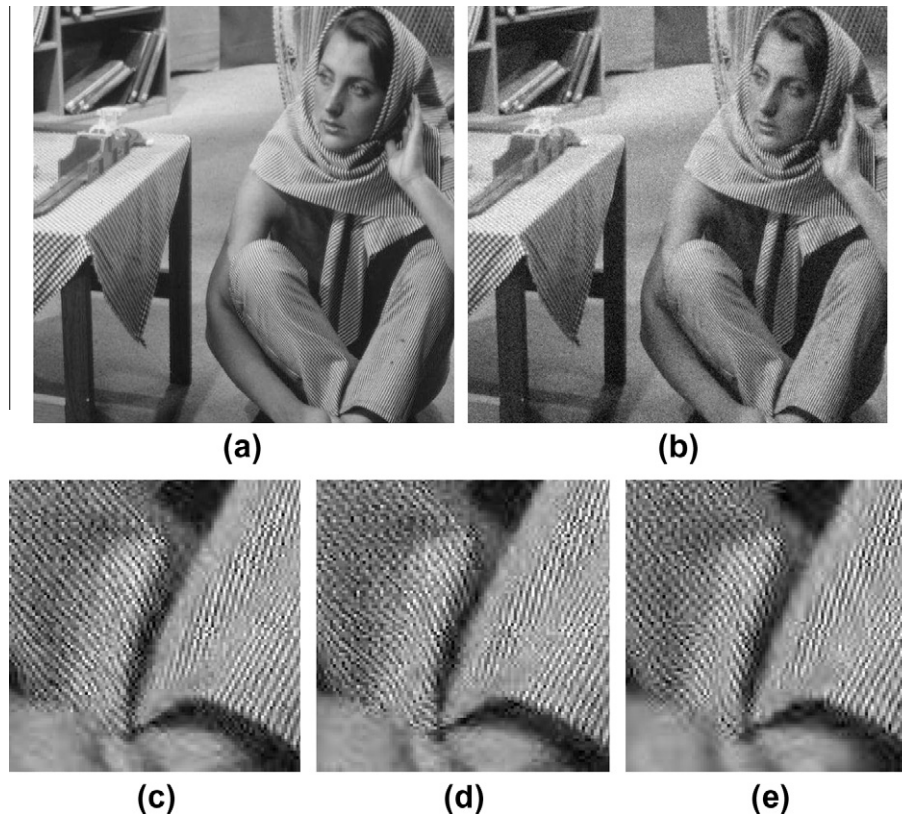
**Fig. 11.** A scale-up details of denoised test image 'Barbara'. The standard variance of noise is 15. (a) Original image. (b) Noisy image, PSNR = 24.61 dB. (c) Ripplet ($c = 1$, $d = 3$) transform, PSNR = 27.55 dB. (d) Curvelet transform, PSNR = 27.24 dB. (e) DWT, PSNR = 27.01 dB.

## 7. Conclusion

To represent edges more efficiently in images, we proposed a new transform called ripplet transform. Our ripplet transform generalizes the existing curvelet transform and is capable of resolving 2D singularities. The nice properties of the ripplet transform are:

- Ripplets form a new tight frame in a function space. The ripplets have good capability of localization in both spatial and frequency domain. The new transform provides a more efficient representation for images or 2D signals.
- The highly directional ripplets have general scaling with arbitrary degree and support, which can capture 2D singularities along different curves in any directions.

Our experimental results showed that the ripplet transform can provide a more efficient representation for images with singularities along smooth curves. When the number of retained coefficients is small, ripplet can outperform DCT and wavelet transform in nonlinear approximation. When applied to image compression, ripplet transform based image coding can achieve roughly 2 dB higher PSNR on average than JPEG, and provide better visual quality than JPEG2000 at low bit-rates.

From our experiments, we noticed that ripplet transform based image coding is suitable for representing texture or edges in images. To further improve the performance of ripplet based image compression, we may combine multiple different transforms so that different regions of an image could be optimally compressed by different transforms. For example, we can apply DCT to smooth areas in an image; for areas with edges or high degree of texture, we can use ripplet transform to preserve more details in these areas. We will pursue this in the future.

The ripplet transform presented in this paper is Type I. The ripplet transform also has Type II and Type III, which are based on parabolic radon transform and cubic radon transform, and we will present Type II and Type III ripplet transform in our future work.

## References

[1] D.L. Donoho, M. Vetterli, R.A. Devore, I. Daubechies, Data compression and harmonic analysis, IEEE Transactions on Information Theory 44 (6) (1998) 2435–2476.
[2] I. Daubechies, Ten Lectures on Wavelets, SIAM, Philadephia, PA, 1992.
[3] S. Mallat, A Wavelet Tour of Signal Processing, second ed., Academic Press, New York, 1999.
[4] E.J. Candes, D.L. Donoho, Ridgelets: a key to higher-dimensional intermittency?, Philosophical Transactions: Mathematical, Physical and Engineering Sciences 357 (1760) (1999) 2495–2509
[5] M. Do, M. Vetterli, The finite Ridgelet transform for image representation, IEEE Transactions on Image Processing 12 (1) (2003) 16–28.
[6] S.R. Deans, The Radon Transform and Some of Its Applications, Wiley, New York, 1983.
[7] J.L. Starck, E.J. Candes, D.L. Donoho, The curvelet transform for image denoising, IEEE Transactions on Image Processing 11 (2002) 670–684.
[8] D.L. Donoho, M.R. Duncan, Digital curvelet transform: strategy, implementation and experiments, in: Proc. of the Aerosense 2000, Wavelet Applications VII. SPIE, vol. 4056, 2000, pp. 12–29.
[9] E. Candes, D. Donoho, Continuous curvelet transform: I. Resolution of the wavefront set, Applied and Computational Harmonic Analysis 19 (2) (2005) 162–197.
[10] E. Candes, D. Donoho, Continuous curvelet transform: II. Discretization and frames, Applied and Computational Harmonic Analysis 19 (2005) 198–222.
[11] L. Hormander, The Analysis of Linear Partial Differential Operators, Springer, Berlin, 2003.
[12] E.J. Candes, D.L. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise $c^2$ singularities, Communications on Pure and Applied Mathematics 57 (2) (2003) 219–266.
[13] M.N. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, IEEE Transactions on Image Processing 14 (12) (2005) 2091–2106.

[14] M.N. Do, M. Vetterli, Contourlets, in: G.V. Welland (Ed.), Beyond wavelets, Academic Press, New York, 2003.

[15] E. Le Pennec, S. Mallat, Sparse geometric image representations with bandelets, IEEE Transactions on Image Processing 14 (4) (2005) 423–438.

[16] D.L.D.E.J. Cands, L. Demanet, L. Ying, Fast discrete curvelet transforms, Multiscale Modeling and Simulation 5 (2005) 861–899.

[17] M. Vetterli, C. Herley, Wavelets and filter banks: theory and design, IEEE Transactions on Signal Processing 40 (9) (1992) 2207–2232.

[18] A. Cohle, I. Daubechies, J. Feauveau, Biothogonal base of compactly supported wavelets, Communications on Pure and Applied Mathematics 4 (1992) 45–47.

[19] M. Manikandan, A. Saravanan, K. Bagan, Curvelet transform based embedded lossy image compression, in: International Conference on Signal Processing, Communications and Networking, ICSCN'07, 2007, pp. 274–276.

[20] D. Taubman, High performance scalable image compression with EBCOT, IEEE Transactions on Image Processing 9 (7) (2000) 1158–1170.

[21] C. Christopoulos, A. Skodras, T. Ebrahimi, The JPEG2000 still image coding system: an overview, IEEE Transactions on Consumer Electronics 46 (4) (2000) 1103–1127.

[22] D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi, I. Ueno, F. Ono, Embedded block coding in JPEG2000, in: Proceedings of the 2000 International Conference on Image Processing, vol. 2, 2000.

[23] Available from: <http://sipi.usc.edu/database/database.cgi?volume=textures>.