

# Ripplet-II Transform for Feature Extraction

Jun Xu and Dapeng Wu<sup>1</sup>

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

---

## Abstract

Current image representation schemes have limited capability of representing 2D singularities (e.g., edges in an image). Wavelet transform has better performance in representing 1D singularities than Fourier transform. Recently invented ridgelet and curvelet transform achieve better performance in resolving 2D singularities than wavelet transform. To further improve the capability of representing 2D singularities, this paper proposes a new transform called ripplet transform Type II (ripplet-II). The new transform is able to capture 2D singularities along a family of curves in images. In fact, ridgelet transform is a special case of ripplet-II transform with degree 1. Ripplet-II transform provide the freedom in parameter settings, which can be optimized for specific problems. Ripplet-II transform can be used for feature extraction due to its efficiency in representing edges and textures. Experiments in texture classification and image retrieval demonstrate that the ripplet-II transform based scheme outperforms wavelet and ridgelet transform based approaches.

*Keywords:* Radon transform, ridgelet transform, wavelet transform, texture classification, image retrieval

---

## 1. Introduction

Efficient representation of images or signals is critical for image processing, computer vision, pattern recognition, and image compression. Harmonic

---

<sup>1</sup>Correspondence author: Prof. Dapeng Wu, wu@ece.ufl.edu, <http://www.wu.ece.ufl.edu>. Some preliminary results in this work have appeared in Ref. [1].

analysis [2] 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 [3, 4]. 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 [5, 6] 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 [7], 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 [8, 9]. Later, they proposed the second generation curvelet transform [10, 11]. 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 [12, 10, 11, 13]. Similar to curvelet, contourlet [14, 15] and bandlet [16] were proposed to resolve 2D singularities.

However, it is not clear why parabolic scaling was chosen for curvelet to achieve anisotropic directionality. To address this, we [17] proposed a new transform called *ripplelet transform* Type I (ripplelet-I), which generalizes the scaling law. Specifically, ripplelet-I transform generalizes curvelet transform by adding two parameters, i.e., support  $c$  and degree  $d$ ; hence, curvelet transform is just a special case of ripplelet-I with  $c = 1$  and  $d = 2$ . The new parameters,

i.e., support  $c$  and degree  $d$ , provide ripplelet-I with anisotropy capability of representing 2D singularities along arbitrarily shaped curves.

Inspired by the success of ridgelet transform, we propose a new transform called *ripplelet transform* Type II (ripplelet-II), which is based on generalized Radon transform [18][19]. The generalized Radon transform converts curves to points. It creates peaks located at the corresponding curve parameters. Intuitively, our ripplelet-II transform consists of two steps: 1) use generalized Radon transform to convert singularities along curves into point singularities in generalized Radon domain; 2) use wavelet transform to resolve point singularities in generalized Radon domain. In this paper, we propose orthogonal ripplelet-II transform. More thorough experimental results with complicated feature extraction are presented in this paper compared to the preliminary results in Ref. [1].

To elaborate, in this paper, we first define the ripplelet-II functions and develop ripplelet-II transform and orthogonal ripplelet-II transform in the continuous space. Then the discrete ripplelet-II transform and orthogonal ripplelet-II transform are defined. Ridgelet transform is just a special case of ripplelet-II transform with degree 1. Properties of ripplelet-II transform are explored and demonstrated by experimental results. Experimental results in texture classification and image retrieval show that ripplelet-II transform provides better feature extraction capability than ridgelet and wavelet based approaches.

The reminder of this paper is organized as below. Section 2 reviews generalized Radon transform. In Section 3, we introduce ripplelet-II transform in both continuous and discrete cases. Section 4 presents the properties of ripplelet-II transform. Experimental results are shown in Section 5 and followed by conclusion in Section 6.

## 2. Generalized Radon Transform

Radon transform is widely applied to tomography [20]. Classical Radon transform is defined in 2D space as the integral of an input 2D function over straight lines. For a 2D integrable real-valued function  $f(x, y)$  where  $(x, y) \in \mathbb{R}^2$ , classical Radon transform of  $f(x, y)$  is defined by

$$R(r, \theta) = \int \int f(x, y) \delta(x \cos \theta + y \sin \theta - r) dx dy \quad (1)$$

Or, we can convert  $f(x, y)$  to  $f(\rho, \phi)$  in polar coordinate system, then classical Radon transform can be calculated by

$$R(r, \theta) = \int \int f(\rho, \phi) \delta(\rho \cos(\phi - \theta) - r) \rho d\rho d\phi \quad (2)$$

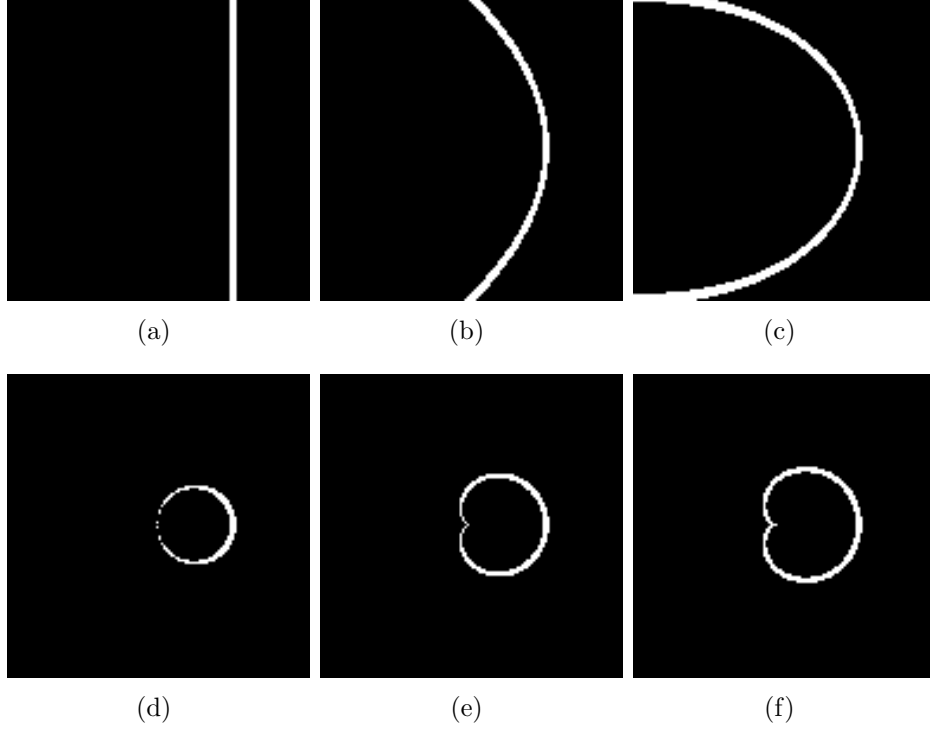


Figure 1: Curves defined by Eq. (3) in Cartesian coordinates. (a)  $d = 1$ . (b)  $d = 2$ . (c)  $d = 3$ . (d)  $d = -1$ . (e)  $d = -2$ . (f)  $d = -3$ .

The classical Radon transform is invertible. The original function can be reconstructed based on the Projection-slice theorem [21]. To extend the classical Radon transform, researchers proposed generalized Radon transform, which is based on an integral along a family of curves [18][19]. In the polar system with coordinates  $(\rho, \phi)$ , a curve can be defined by

$$\rho^{1/d} \cos\left(\frac{1}{d}(\phi - \theta)\right) = r^{1/d} \quad (3)$$

where  $r$  and  $\theta$  are fixed, and  $d$  denotes degree. For  $d = 1$  and  $d = 2$ , Eq. (3) represents straight line and parabola as shown in Figure 1(a) and 1(b), respectively. For  $d = -1$  and  $d = -2$ , Eq. (3) represents circles through the

origin and Cardioids as shown in Figure 1(d) and 1(e), respectively. When  $0 < d < 1$  or  $-1 < d < 0$ , curves intersect themselves at least once. Then we have a single curve. Otherwise, curves do not intersect themselves, which will lead to complicate situations. So we only consider  $|b| \geq 1$ . We refer  $d > 0$  as ‘positive curves’ and  $d < 0$  as ‘negative curves’ in the rest of this paper.

The generalized Radon transform along curves can be defined in the polar coordinates  $(\rho, \phi)$  by

$$GR_d(r, \theta) = \int \int \rho f(\rho, \phi) \delta(r - \rho \cos^d((\phi - \theta)/d)) d\rho d\phi \quad (4)$$

If  $d = 1$ , Eq. (4) reduces to the classical Radon transform in Eq. (2). If  $d = 2$ , Eq. (4) becomes [22]

$$\begin{aligned} GR_2(r, \theta) &= 2\sqrt{r} \int \int \rho' f(\rho'^2, 2\phi') \delta(\rho' \cos(\phi' - \frac{1}{2}\theta) - \sqrt{r}) d\phi' d\rho' \\ &= 2\sqrt{r} R[f(\rho'^2, 2\phi')](\sqrt{r}, \theta/2) \end{aligned} \quad (5)$$

where  $R[f(x, y)](r, \theta)$  denotes the classical Radon transform that maps  $f(x, y)$  to  $R(r, \theta)$ , and is defined in Eq. (1); note that  $f(\rho, \phi)$  under the polar coordinate system needs to be converted to  $f(x, y)$  under Cartesian coordinate system before computing Eq. (1). Eq. (5) shows that for  $d = 2$ , the generalized Radon transform can be implemented via the classical Radon transform with appropriate substitutions of variables.

For the general case, i.e.,  $d \in \mathbb{Z}$ , the generalized Radon transform can be computed via Fourier series [18][19]. Let  $f(\rho, \phi)$  be a 2D function defined in polar coordinates  $(\rho, \phi)$  and  $GR_d(r, \theta)$  be its generalized Radon transform. Assume that the Fourier series for  $f(\rho, \phi)$  exists, i.e.,

$$f(\rho, \phi) = \sum_{n=-\infty}^{+\infty} f_n(\rho) e^{in\phi} \quad (6)$$

where

$$f_n(\rho) = \int f(\rho, \phi) e^{-in\phi} d\phi \quad (7)$$

Then the generalized Radon transform can be computed by

$$GR_d(r, \theta) = \sum_{n=-\infty}^{+\infty} g_n(r) e^{in\theta} \quad (8)$$

where for  $d > 0$

$$\begin{aligned} g_n(r) &= 2 \int_r^\infty f_n(\rho) \frac{\cos\{(nd) \cos^{-1}(r/\rho)^{1/d}\}}{(1 - (r/\rho)^{2/d})^{1/2}} d\rho \\ &= 2 \int_r^\infty f_n(\rho) \times (1 - (r/\rho)^{2/d})^{-1/2} \times T_{nd}((r/\rho)^{1/d}) d\rho \end{aligned} \quad (9)$$

where  $T_n(\cdot)$  is the Chebyshev polynomial of degree  $n$ , and  $f_n(\rho)$  is given by Eq. (7).

Putting Eqs. (6), (7), (8), (9) together, we have the generalized Radon transform of the function  $f$  as

$$GR_d(r, \theta) = 2 \sum_{n=-\infty}^{+\infty} \left[ \int_r^\infty \int f(\rho, \phi) e^{-in\phi} d\phi \times (1 - (r/\rho)^{2/d})^{-1/2} \times T_{nd}((r/\rho)^{1/d}) d\rho \right] e^{in\theta} \quad (10)$$

The inverse transform is defined by

$$\begin{aligned} f(\rho, \phi) &= -\frac{1}{\pi} \sum_{n=-\infty}^{+\infty} \left[ \frac{d}{d\rho} \int_\rho^\infty \int_0^{2\pi} GR_d(r, \theta) e^{-in\theta} \times T_{nd}((r/\rho)^{1/d}) \right. \\ &\quad \left. \times ((r/\rho)^{2/d} - 1)^{-1/2} \frac{1/d}{r} dr d\theta \right] e^{in\phi} \end{aligned} \quad (11)$$

For negative curves (i.e.  $d < 0$ ), the Generalized Radon transform is

$$GR_d(r, \theta) = 2 \sum_{n=-\infty}^{+\infty} \left[ \int_0^r \int f(\rho, \phi) e^{-in\phi} d\phi \times (1 - (r/\rho)^{-2/d})^{-1/2} \times T_{-nd}((r/\rho)^{-1/d}) d\rho \right] e^{in\theta} \quad (12)$$

The inverse transform for negative curves is defined by

$$\begin{aligned} f(\rho, \phi) &= \frac{1}{\pi} \sum_{n=-\infty}^{+\infty} \left[ \frac{d}{d\rho} \int_0^\rho \int_0^{2\pi} GR_d(r, \theta) e^{-in\theta} \times T_{-nd}((r/\rho)^{-1/d}) \right. \\ &\quad \left. \times ((r/\rho)^{-2/d} - 1)^{-1/2} \frac{1/d}{r} dr d\theta \right] e^{in\phi} \end{aligned} \quad (13)$$

### 3. Ripplet-II transform

#### 3.1. Continuous Ripplet-II Transform

To present ripplet-II transform, we need to define ripplet-II functions first. Given a smooth univariate wavelet function  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$  with  $\int \varphi(t)dt = 0$ , we define a bivariate function  $\psi_{a,b,d,\theta} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  in the polar coordinate system by

$$\psi_{a,b,d,\theta}(\rho, \phi) = a^{-1/2} \varphi((\rho \cos^d((\theta - \phi)/d) - b)/a) \quad (14)$$

where  $a > 0$  denotes scale,  $b \in \mathbb{R}$  denotes translation,  $d \in \mathbb{N}$  denotes degree, and  $\theta \in [0, 2\pi)$  denotes orientation. Function  $\psi_{a,b,d,\theta}$  is called ripplet-II function. Here, we only consider  $d > 0$  (i.e. positive curves), since positive curves are open curves. Examples of ripplet-II functions with different parameter settings are shown in Figure 2. Ripplet-II can be scaled, translated and rotated according to the parameters  $a, b, \theta$ . Note that when  $d = 1$ , ripplet-II reduces to ridgelet as shown in Figure 3; i.e., ridgelet transform is just a special case of ripplet-II transform with  $d = 1$ .

#### Forward Transform

Ripplet-II transform of a real-valued 2D function  $f$  is defined as the inner product between the function  $f$  and ripplet-II functions

$$\mathcal{R}_f(a, b, d, \theta) = \int \int \bar{\psi}_{a,b,d,\theta}(\rho, \phi) f(\rho, \phi) \rho d\rho d\phi \quad (15)$$

where  $\bar{\psi}$  is the complex conjugate of  $\psi$ , and  $f(\rho, \phi)$  is under the polar coordinate system.

Ripplet-II transform has the capability of capturing structure information along arbitrary curves by tuning the scale, location, orientation, and degree parameters. From Eq. (15), we have

$$\begin{aligned} \mathcal{R}_f(a, b, d, \theta) &\stackrel{(a)}{=} \int \int \bar{\psi}_{a,b,d,\theta}(\rho, \phi) f(\rho, \phi) \rho d\rho d\phi \\ &\stackrel{(b)}{=} \int \int a^{-1/2} \bar{\varphi}((\rho \cos^d((\theta - \phi)/d) - b)/a) f(\rho, \phi) \rho d\rho d\phi \\ &= \int \int a^{-1/2} \int \bar{\varphi}((r - b)/a) \delta(r - \rho \cos^d((\theta - \phi)/d)) dr f(\rho, \phi) \rho d\rho d\phi \\ &= \int a^{-1/2} \bar{\varphi}((r - b)/a) \times \left[ \int \int \delta(r - \rho \cos^d((\theta - \phi)/d)) f(\rho, \phi) \rho d\rho d\phi \right] dr \\ &\stackrel{(c)}{=} \langle \varphi_{a,b}, GR_d[f] \rangle \end{aligned} \quad (16)$$

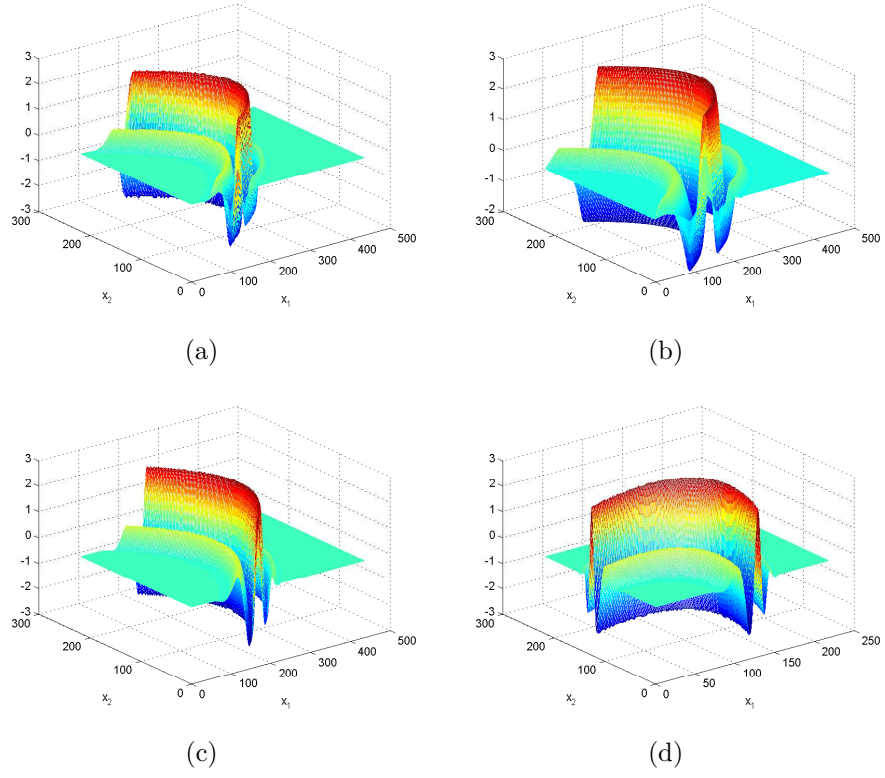
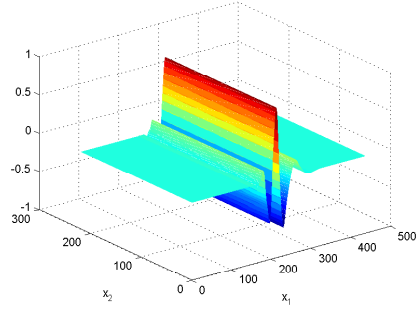
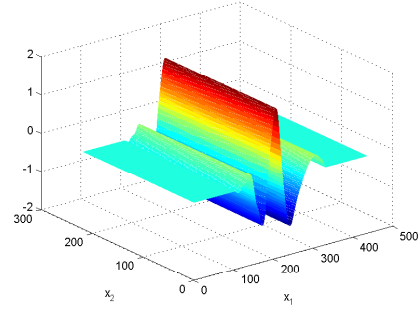


Figure 2: Ripplet-II functions in Cartesian coordinates  $(x_1, x_2)$  (a)  $a = 1, b = 0, d = 2$  and  $\theta = 0$ . (b)  $a = 2, b = 0, d = 2$  and  $\theta = 0$ . (c)  $a = 1, b = 0.05, d = 2$  and  $\theta = 0$ . (d)  $a = 1, b = 0, d = 2$  and  $\theta = 30$ .

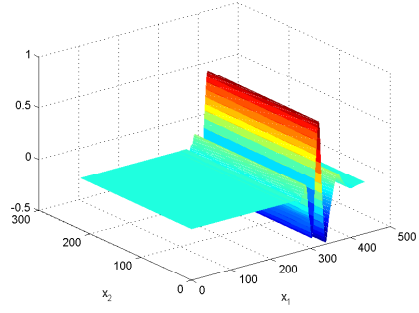




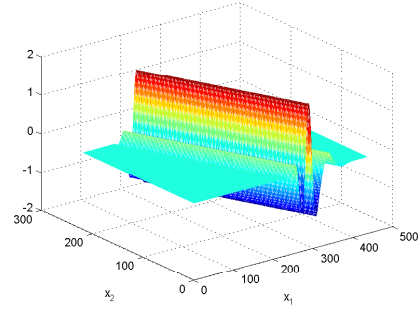
(a)



(b)



(c)



(d)

Figure 3: Ripplet-II functions in Cartesian coordinates  $(x_1, x_2)$  (a)  $a = 1, b = 0, d = 1$  and  $\theta = 0$ . (b)  $a = 2, b = 0, d = 1$  and  $\theta = 0$ . (c)  $a = 1, b = 0.05, d = 1$  and  $\theta = 0$ . (d)  $a = 1, b = 0, d = 1$  and  $\theta = 30$ .

where (a) is due to Eq. (15); (b) is due to Eq. (14); (c) is due to Eq. (4) and  $GR_d[f]$  is the generalized Radon transform (GRT) of function  $f$ . Eq. (16) shows that ripplelet-II transform can be obtained by the inner product between GRT and 1D wavelet, which is the 1D wavelet transform (WT) of GRT of function  $f$ ; i.e., the ripplelet-II transform of function  $f$  can be obtained by first computing GRT of  $f$ , and then computing 1D WT of the GRT of  $f$  as below:

$$f(\rho, \phi) \xrightarrow{GRT} GR_d[f](r, \theta) \xrightarrow{1D-WT} \mathcal{R}_f(a, b, d, \theta) \quad (17)$$

where the 1D WT is with respect to (w.r.t.)  $r$ .

In details, ripplelet-II transform of  $f$  can also be obtained through

$$\begin{aligned} \mathcal{R}_f(a, b, d, \theta) = 2 \sum_{n=-\infty}^{+\infty} \int a^{-1/2} \bar{\varphi}((r-b)/a) \int_r^\infty \int f(\rho, \phi) e^{-in\phi} d\phi \\ \times (1 - (r/\rho)^{2/d})^{-1/2} \times T_{nd}((r/\rho)^{1/d}) d\rho e^{in\theta} dr \end{aligned} \quad (18)$$

### Inverse Transform

Ripplelet-II transform is invertible. Given ripplelet-II coefficients  $\mathcal{R}_f(a, b, d, \theta)$ , we can reconstruct the original function  $f$  through

$$\begin{aligned} f(\rho, \phi) = -\frac{1}{\pi} \sum_{n=-\infty}^{+\infty} \left[ \frac{d}{d\rho} \int_\rho^\infty \int_0^{2\pi} \int_0^\infty \int_{-\infty}^\infty \frac{1}{\sqrt{a}} \mathcal{R}_f(a, b, d, \theta) \varphi\left(\frac{r-a}{b}\right) e^{-in\theta} \right. \\ \left. \times T_{nd}((r/\rho)^{1/d}) \times ((r/\rho)^{2/d} - 1)^{-1/2} \frac{1/d}{r} da db dr d\theta \right] e^{in\phi} \end{aligned} \quad (19)$$

Reversing the process in (17), the inverse of the ripplelet-II transform of function  $f$  can be obtained by first computing inverse WT (IWT) of  $\mathcal{R}_f(a, b, d, \theta)$  w.r.t.  $a$  and  $b$ , and then computing inverse GRT (IGRT) as below:

$$\mathcal{R}_f(a, b, d, \theta) \xrightarrow{1D-IWT} GR_d[f](r, \theta) \xrightarrow{IGRT} f(\rho, \phi) \quad (20)$$

where IGRT can be computed by the method in Section 2, Eq. (11).

### 3.2. Continuous Orthogonal Ripplelet-II Transform

As shown in (17), ripplelet-II transform can be implemented as a 1D wavelet transform along the radius of the generalized Radon domain. If we apply

2D wavelet transform to the generalized Radon coefficients, the additional wavelet transform along angle  $\theta$  holds the potential of improving the sparsity of transform coefficients. We call the new extension *orthogonal ripplelet-II* transform.

In mathematics, orthogonal ripplelet-II transform of a function  $f(\rho, \phi)$  in polar coordinates is defined by

$$\begin{aligned} \mathcal{R}_f^{orth}(a, b_1, b_2, d) = 2 \sum_{n=-\infty}^{+\infty} \int \int \frac{1}{a} \bar{\varphi}\left(\frac{r-b_1}{a}\right) \bar{\varphi}\left(\frac{\theta-b_2}{a}\right) \int_r^\infty \int f(\rho, \phi) e^{-in\phi} d\phi \\ \times (1 - (r/\rho)^{2/d})^{-1/2} T_{nd}((r/\rho)^{1/d}) d\rho e^{in\theta} dr d\theta \end{aligned} \quad (21)$$

Similar to ripplelet-II transform, orthogonal ripplelet-II transform of the function  $f$  can be obtained by first computing GRT of  $f$ , and then computing 2D WT of the GRT of  $f$  as below:

$$f(\rho, \phi) \xrightarrow{GRT} GR_d[f](r, \theta) \xrightarrow{2D-WT} \mathcal{R}_f^{orth}(a, b_1, b_2, d) \quad (22)$$

There is no direction parameter in orthogonal ripplelet-II coefficients  $\mathcal{R}_f^{orth}(a, b_1, b_2, d)$ . This may not provide explicit information about the directions of curves. However, due to the additional wavelet transform along angles, sparser representation of functions is achieved.

Orthogonal ripplelet-II transform is also invertible. Given orthogonal ripplelet-II coefficients  $\mathcal{R}_f^{orth}(a, b_1, b_2, d)$ , we can reconstruct the original function  $f$  through

$$\begin{aligned} f(\rho, \phi) = -\frac{1}{\pi} \sum_{n=-\infty}^{+\infty} \left[ \frac{d}{d\rho} \int_\rho^\infty \int_0^{2\pi} \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \frac{1}{a} \mathcal{R}_f^{orth}(a, b_1, b_2, d) \varphi\left(\frac{r-a}{b_1}\right) \varphi\left(\frac{\theta-a}{b_2}\right) e^{-in\theta} \right. \\ \left. \times T_{nd}((r/\rho)^{1/d}) \times ((r/\rho)^{2/d} - 1)^{-1/2} \frac{1/d}{r} da db_1 db_2 dr d\theta \right] e^{in\phi} \end{aligned} \quad (23)$$

Reversing the process in (22), the inverse of the orthogonal ripplelet-II transform of function  $f$  can be obtained by first computing inverse 2D WT (IWT) of  $\mathcal{R}_f^{orth}(a, b_1, b_2, d)$  w.r.t.  $a$ ,  $b_1$  and  $b_2$ , and then computing inverse GRT (IGRT) as below:

$$\mathcal{R}_f^{orth}(a, b_1, b_2, d) \xrightarrow{2D-IWT} GR_d[f](r, \theta) \xrightarrow{IGRT} f(\rho, \phi) \quad (24)$$

### 3.3. Discrete Ripplet-II Transform

If the input of ripplet-II transform is a digital image, we need to use discrete ripplet-II transform. Following the paradigm in (17), discrete Ripplet-II transform of function  $f$  can be obtained by first computing discrete GRT (DGRT) of  $f$ , and then computing 1D discrete WT (DWT) of the DGRT of  $f$  as below:

$$f(\rho, \phi) \xrightarrow{DGRT} GR_d[f](r, \theta) \xrightarrow{1D-DWT} \mathcal{R}_f(a, b, d, \theta) \quad (25)$$

The discrete orthogonal ripplet-II transform follows the paradigm in (22) and is obtained by

$$f(\rho, \phi) \xrightarrow{DGRT} GR_d[f](r, \theta) \xrightarrow{2D-DWT} \mathcal{R}_f^{orth}(a, b_1, b_2, d) \quad (26)$$

If  $d = 2$ , there is a simpler method to computer discrete ripplet-II transform, the details of which will be elaborated in Section 3.4.

### 3.4. Discrete Ripplet-II Transform with $d = 2$

For  $d = 2$ , the generalized Radon transform is called *parabolic Radon transform* [22]. Eq. (5) shows that for  $d = 2$ , the generalized Radon transform can be implemented via the classical Radon transform with appropriate substitutions of variables. Hence, we can compute discrete ripplet-II transform via Eqs. (25) and (5).

*Computation of Forward Ripplet-II Transform with  $d = 2$ .* The forward transform can be obtained by the following steps.

1. Convert Cartesian coordinates to polar coordinates, i.e., convert  $f(x, y)$  to  $f(\rho, \phi)$ . For  $f(\rho, \phi)$ , substitute  $(\rho, \phi)$  with  $(\rho'^2, 2\phi')$ . Convert polar coordinates  $(\rho', \phi')$  to Cartesian coordinates  $(x, y)$ , and obtain new image  $f_1(x, y)$  by interpolation, where  $x$  and  $y$  are integer-valued.
2. Apply classical Radon transform to  $f_1(x, y)$ , resulting in  $R(r', \theta')$ . In function  $R(r', \theta')$ , substitute  $(r', \theta')$  with  $(\sqrt{r}, \theta/2)$ ; and obtain the generalized Radon coefficients  $GR_2(r, \theta)$  via Eq. (5).
3. Apply 1D wavelet transform to  $GR_2(r, \theta)$  with respect to  $r$ , and obtain the ripplet-II coefficients.

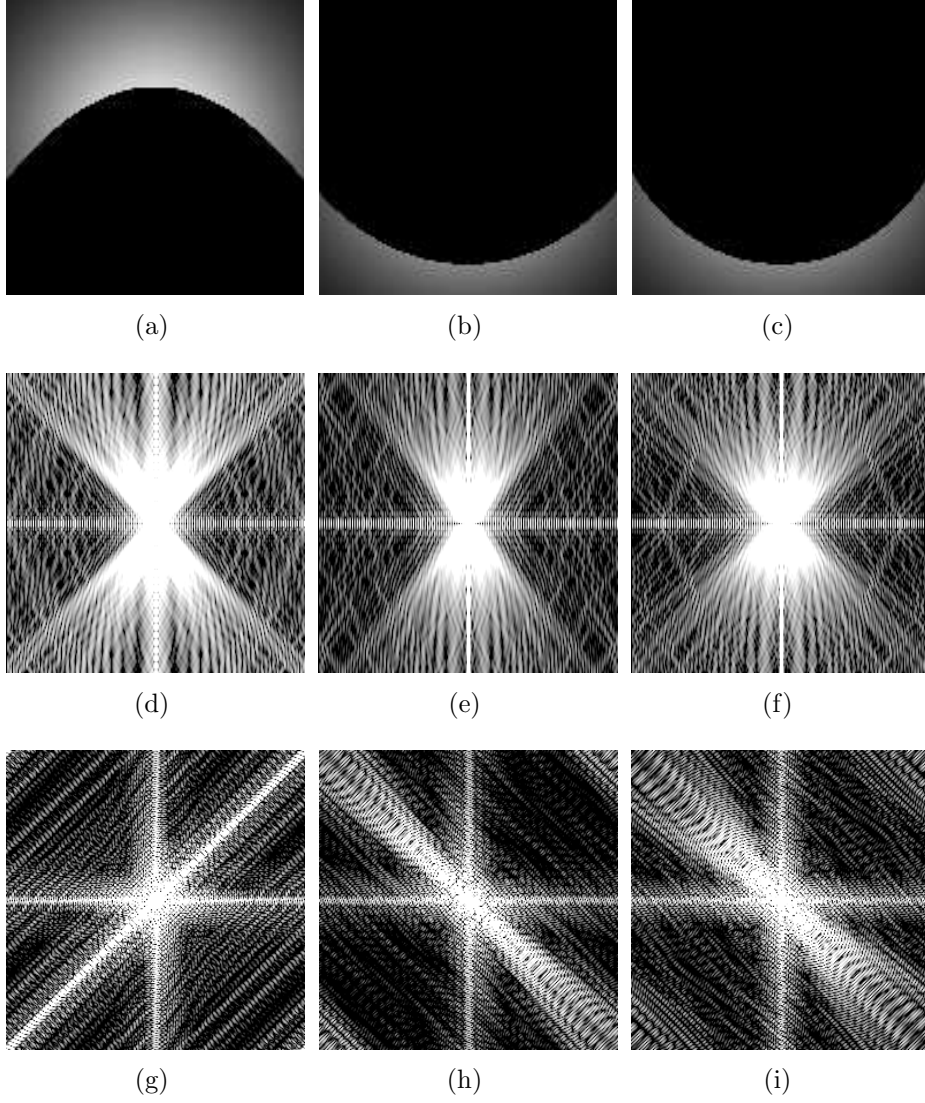


Figure 4: Gaussian images with a curve edge. Top row: original image  $f(x, y)$ ; Middle row: Magnitude of 2D Fourier transform; Bottom row: Magnitude of 2D Fourier transform after substituting the polar coordinate  $(r', \theta')$  with  $(\sqrt{r}, \theta/2)$ . Left column: Parabolic curve. Middle column: Curve with degree 3. Right column: Curve with degree 4

To show the sparsity of ripplelet-II transform with  $d = 2$ , we plot Figure 4. As we know, ridgelet transform of a 2D function  $f(x, y)$  is computed by 1) 2D Fourier transform of  $f(x, y)$ , 2) converting Cartesian coordinate system to the polar coordinate system  $(\omega', \theta')$ , 3) 1D inverse Fourier transform w.r.t.  $\omega'$ , resulting in  $(r', \theta')$ , 4) 1D wavelet transform w.r.t.  $r'$ . In contrast, ripplelet-II transform of a 2D function  $f(x, y)$  is computed by 1) 2D Fourier transform of  $f_1(x, y)$ , 2) converting Cartesian coordinate system to the polar coordinate system  $(\omega', \theta')$ , 3) 1D inverse Fourier transform w.r.t.  $\omega'$ , resulting in  $(r', \theta')$ , 4) substituting  $(r', \theta')$  with  $(\sqrt{r}, \theta/2)$ , 5) 1D wavelet transform w.r.t.  $r$ . The key different between ripplelet-II transform and ridgelet transform in their computing procedures is that ripplelet-II transform has an extra step, i.e., substituting  $(r', \theta')$  with  $(\sqrt{r}, \theta/2)$ . If we apply 1D wavelet transform to the middle row in Figure 4, we obtain ridgelet transform coefficients. It is observed that the Fourier transform coefficients in the bottom row of Figure 4 are sparser than those in the middle row of Figure 4. This is why ripplelet-II transform provides sparser coefficients than ridgelet transform. In other words, substituting  $(r', \theta')$  with  $(\sqrt{r}, \theta/2)$  helps make coefficients sparser.

*Computation of Inverse Ripplelet-II Transform with  $d = 2$ .* The inverse transform can be obtained by the following steps.

1. Apply 1D inverse wavelet transform to ripplelet-II coefficients with respect to  $r$ , resulting in  $GR_2(r, \theta)$ .
2. In function  $GR_2(r, \theta)/(2\sqrt{r})$ , substitute  $(r, \theta)$  with  $(r'^2, 2\theta')$ , resulting in  $R(r', \theta')$ .
3. Apply classical inverse Radon transform to  $R(r', \theta')$ , resulting in  $f_1(x, y)$ .
4. For  $f_1(x, y)$ , convert Cartesian coordinates  $(x, y)$  to polar coordinates  $(\rho', \phi')$ , resulting in  $f_1(\rho', \phi')$ . For function  $f_1(\rho', \phi')$ , substitute  $(\rho', \phi')$  with  $(\sqrt{\rho}, \phi/2)$ , resulting in  $f(\rho, \phi)$ . For  $f(\rho, \phi)$ , convert polar coordinates  $(\rho, \phi)$  to Cartesian coordinates  $(x, y)$ , and obtain  $f(x, y)$  by interpolation, where  $x$  and  $y$  are integer-valued.

*Computation of Orthogonal Ripplelet-II Transform with  $d = 2$ .* The computation of orthogonal ripplelet-II transform is similar to that of ripplelet-II transform. The only difference is to replace 1D wavelet transform with 2D wavelet transform. Particularly, the forward orthogonal ripplelet-II transform is implemented by first computing Step 1 and 2, and then replacing 1D wavelet transform to 2D wavelet transform in Step3. The inverse orthogonal ripplelet-II transform is computed by first replacing 1D inverse wavelet transform to

2D inverse wavelet transform in Step 1, and then following the remaining steps.

#### 4. Properties of Ripplet-II Transform

According to the definition, we can directly find the following properties about ripplet-II transform

- Localization: Ripplet-II with degree  $d$  decays fast along curves of polynomial degree  $d$ .
- Directionality: Ripplet-II can be oriented toward arbitrary direction.
- Flexibility: Compared to ridgelet, ripplet-II provide flexible choice for degrees. Optimal degree can be determined for specific applications.

In Figure 5, ripplet-II representation of images with different degrees are presented. It can be observed that different degrees present different levels of sparsity.

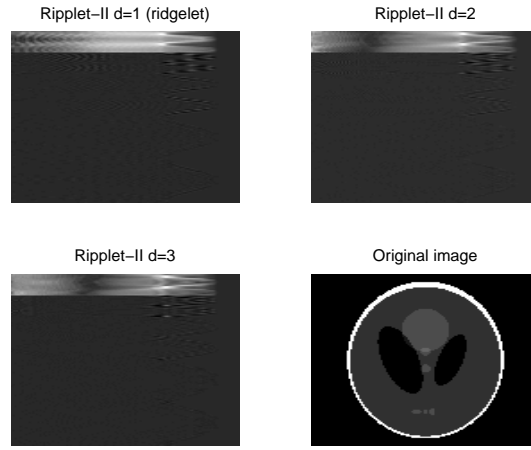
Figure 6 shows the magnitude of transform coefficients in a decreasing order for wavelet, ridgelet, ripplet-II and orthogonal ripplet-II transforms; the magnitude of coefficients of each transform is normalized by the largest coefficient of the corresponding transform. It can be observed that ripplet-II has the fastest decay in coefficients, compared to wavelet and ridgelet. This is the reason why ripplet-II transform can provide sparser representation for images with edges than wavelet and ridgelet. In Figure 6, orthogonal ripplet-II demonstrates faster decay than ripplet-II, which indicates that orthogonal ripplet-II transform can provide sparser representation of functions than ripplet-II as stated in Section 3.2.

Besides the aforementioned properties, ripplet-II transform can provide rotation invariance. We show this as below. If we have an image  $f_1(x, y)$  as well as its rotated version  $f_2(x, y)$  rotated by an angle  $\alpha$ , i.e.,

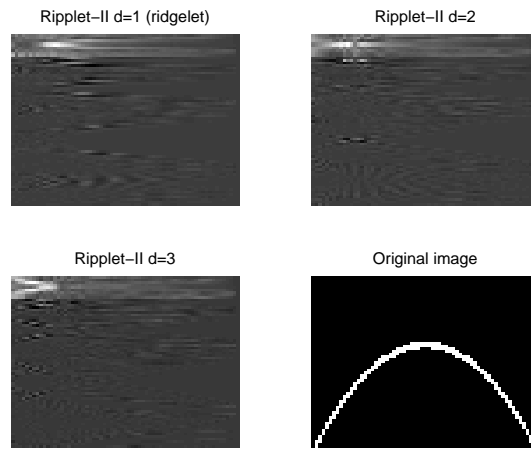
$$f_2(x, y) = f_1(x \cos(\alpha) + y \sin(\alpha), -x \sin(\alpha) + y \cos(\alpha)) \quad (27)$$

In the polar coordinate system, we have

$$f_2(\rho, \phi) = f_1(\rho, \phi - \alpha) \quad (28)$$



(a) Phantom



(b) Parabola

Figure 5: Ripplet-II with different degrees.



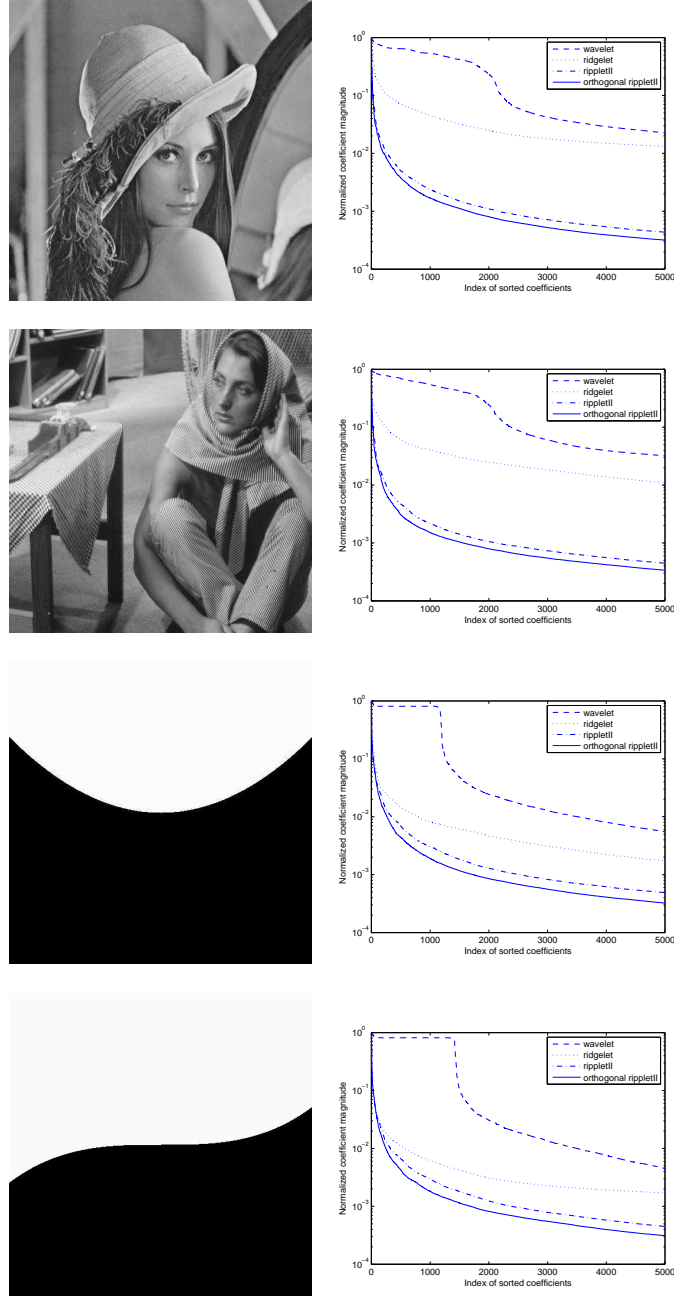


Figure 6: Comparison of coefficient decaying between wavelet, ridgelet, ripplelet-II and orthogonal ripplelet-II with  $d = 2$ . Left column: original images. Right column: coefficient decaying comparison.

So ripplelet-II transform of  $f_2$  is

$$\begin{aligned}
\mathcal{R}_{f_2}(a, b, d, \theta) &= \int \int \bar{\psi}_{a,b,d,\theta}(\rho, \phi) f_2(\rho, \phi) \rho d\rho d\phi \\
&= \int \int \bar{\psi}_{a,b,d,\theta}(\rho, \phi) f_1(\rho, \phi - \alpha) \rho d\rho d\phi \\
&= \int \int \bar{\psi}_{a,b,d,\theta-\alpha}(\rho, \phi) f_1(\rho, \phi) \rho d\rho d\phi \\
&= \mathcal{R}_{f_1}(a, b, d, \theta - \alpha)
\end{aligned} \tag{29}$$

Applying 1D Fourier transform on both sides of Eq. (29) with respect to  $\theta$ , we have

$$F\{\mathcal{R}_{f_2}(a, b, d, \theta)\} = \int \mathcal{R}_{f_1}(a, b, d, \theta - \alpha) e^{-i\omega\theta} d\theta = e^{-i\omega\alpha} F\{\mathcal{R}_{f_1}(a, b, d, \theta)\} \tag{30}$$

Obviously, we have  $|F\{\mathcal{R}_{f_1}(a, b, d, \theta)\}| = |F\{\mathcal{R}_{f_2}(a, b, d, \theta)\}|$ ; i.e., the magnitude of 1D Fourier transform (w.r.t.  $\theta$ ) of ripplelet-II transform is rotation invariant. Hence, ripplelet-II transform can provide rotation invariant features. Since there is no explicit direction parameter in orthogonal ripplelet-II coefficient, orthogonal ripplelet-II transform does not have the rotation invariant property.

## 5. Experimental Results

In this section, we evaluate the performance of ripplelet-II transform in the problems of texture classification and image retrieval, where ripplelet-II transform serves as a feature extractor. Our experiments use the texture volume in USC-SIPI image database [23]. The texture volume consists of 2 sub-databases, all of which contain monochrome texture images.

### 5.1. Texture Classification

This experiment is to compare ripplelet-II with other transforms as feature extraction tools in the scenario of texture classification. A sub-database named *Rotated Textures* [24] in the texture volume contains a set of rotated textures. Each image in the sub-database is of size 512×152 pixels. The sub-database contains a total of 13 textures as shown in Figure 7 and each texture has 7 versions, which are rotated by 0°, 30°, 60°, 90°, 120°, 150°, and 200°. Hence, the sub-database contains a total of  $13 \times 7 = 91$  images.

Given a image from the sub-database, the algorithm will return a label that indicates which texture the image belongs to. Next, we describe the feature extraction and classification algorithms used in the experiments.

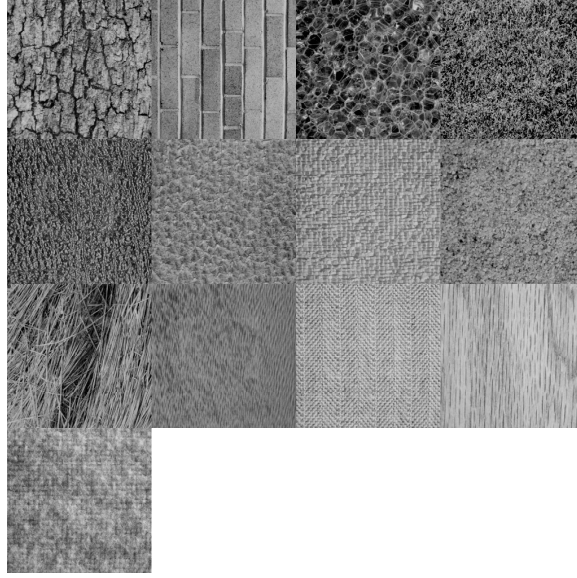


Figure 7: Textures used in texture classification.

Table 1: Information extracted from data

Notation	Equation	Description
C1	$\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M  R(i, j) $	first absolute moment
C2	$\frac{1}{NM-1} \sum_{i=1}^N \sum_{j=1}^M  R(i, j) - \bar{R}(i, j) ^2$	variance
C3	$\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M  R(i, j) ^2$	average energy
C4	$-\sum_{i=1}^k P_i \log P_i$	entropy

*Feature extractor 1.* We first partition the transform coefficient matrix into  $N_D$  nonoverlapped blocks. So the feature dimension is  $N_D$ . The feature extracted from transform domains is the statistical information from nonoverlapped blocks of size  $N \times M$ . Information such as first absolute moment, variance, average energy and entropy carries geometry information from the

Table 2: Error rate under different transforms using feature extractor 1

Feature dimension $N_D$	C1			
	Wavelet	Ridgelet	Ripplet-II	Orth-Ripplet-II
1	0.4885	0.9341	0.2802	0.1099
4	0.0441	0.8462	0.0019	0.1319
16	0.0187	0.7912	0	0.0879
64	0.0057	0.8462	0	0.0879
256	0.0024	0.9121	0	0.0879
	C2			
1	0.2826	0.4286	0.1715	0.1099
4	0.0929	0.3516	0.0728	0.1099
16	0.0536	0.4176	0.0608	0.1319
64	0.0374	0.5385	0.0580	0.1429
256	0.0172	0.5824	0.0656	0.1099
	C3			
1	0.2883	0.4286	0.1844	0.1209
4	0.0072	0.3077	0.0024	0.1319
16	0.0010	0.3846	0.0010	0.1099
64	0.0010	0.4835	0.0010	0.1099
256	0.0019	0.5055	0	0.0879
	C4			
1	0.6624	0.8352	0.4061	0.5824
4	0.8841	0.8352	0.1049	0.3407
16	0.9176	0.7143	0.0532	0.2967
64	0.9746	0.6923	0.2055	0.2967
256	0.9909	0.6374	0.1173	0.2857

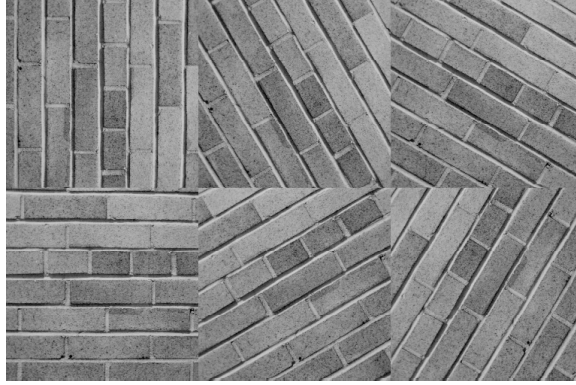


Figure 8: Textures rotated with different angles.

original image as shown in Table 1 [25]. The rotation invariant property of ripplelet-II transform guarantees that rotated images have almost the same feature vector as that of the original image. The statistical information from Table 1 can also provide rotation invariant property for wavelet transform approach.

*Feature extractor 2.* We apply a transform to each image and obtain a vector of transform coefficients. Assume that we have  $N_t$  images for training. Then, we have  $N_t$  vectors of transform coefficients, which form a matrix. We apply principle component analysis (PCA) [26] to this transform coefficient matrix and obtain eigenvalues/eigenvectors of the matrix. PCA provides a transformation matrix, which consists of principle components (normalized eigenvectors); we multiply the transformation matrix and a transform coefficient vector, resulting in a feature vector. We choose the feature dimensions that corresponds to the  $N_D$  principal components whose eigenvalues are largest. Hence, the resulting feature vector is  $N_D$ -dimensional.

*Classification algorithm.* We use k-nearest-neighbor (kNN) classifier where  $k = 5$ . The distance measure used in kNN is  $N_D$ -dimensional Euclidean distance. A leave-one-out cross-validation classification algorithm is used to evaluate the classification performance. Specifically, we first compute the distance between a test feature vector and each of the feature vectors with known labels, and then determine the class label of the test feature vector by a k-nearest-neighbor classifier. Our performance measure is error rate, which is the ratio of the number of mis-classified images to the total number of images tested.

We test four types of transforms, i.e., ridgelet, wavelet, ripplelet-II and orthogonal ripplelet-II transform. Ridgelet transform implementation is available [27] and default settings are used. Wavelet transform uses 'Daubechies 4'(db4) wavelets. Ripplelet-II with degree 2 and orthogonal ripplelet-II with degree 2 are used in this experiment. Texture classification performances are listed in Table 2 and Table 3 corresponding to feature extractor 1 and feature extractor 2, respectively. Table 2 shows that ripplelet-II transform achieves lower error rate than wavelet transform under all feature vector lengths tested and under C1, C3, and C4; ripplelet-II transform achieves higher error rate than wavelet transform under C2 for feature vector length larger than or equal to 16. Results also indicate that the entropy method C4 is not a good statistical feature. Table 2 shows that ripplelet-II transform outperforms other transforms in almost all cases by achieving lowest error rate. However, orthogonal ripplelet-II transform is worse than ripplelet-II and wavelet. Table 3 shows that ripplelet-II transform achieves lower error rate than ridgelet, wavelet and orthogonal ripplelet-II transform under all feature dimensions tested. The reason why ripplelet-II transform achieves the best classification performance is two-folded. First, ripplelet-II transform is able to provide sparser feature vectors than ridgelet and wavelet transform. Second, the rotation invariant property of ripplelet-II transform guarantees that rotated images have almost the same feature vector as that of the original image.

Table 3: Error rate under different transforms using feature extractor 2

Feature dimension $N_D$	Ridgelet	Wavelet	Ripplelet-II	Orth-Ripplelet-II
1	0.3908	0.2639	0.1341	0.1758
2	0.2648	0.0752	0.0172	0.1538
4	0.2261	0.0527	0.0029	0.1538
8	0.1676	0.0350	0.0005	0.1319
16	0.1058	0.0177	0	0.1319
32	0.0704	0.0105	0	0.1319

In Table 4, we present the best performance of various transform based approaches on the sub-database with 91 images. Gabor wavelet [28, 29] is used for feature extraction. The frequency range is [0.05, 0.4] and Gabor

Table 4: Error rate comparisons

Transform	Best error rate
Ridgelet	0.07
Gabor Wavelet	0.11
Ripplet-I	0
Ripplet-II	0

envelop is calculated accordingly [28, 29]. Gabor wavelet filters with 5 scales and 6 directions are applied to obtain 30 subbands coefficients of an image. The feature vector consists of the mean and standard derivation of each subband as [30]. Then the feature dimension is 60. When we apply ripplet-I [17] to images, we will have ripplet-I coefficients with different scales and different directions. Due to the property of ripplet-I, the numbers of directions at each scale are different. The feature vector of ripplet-I consists of the mean and standard derivation of each subband from the ripplet-I coefficients. In this experiment, the dimension of ripplet-I feature is 38.

Results in Table 4 show that both ripplet-I and ripplet-II achieve better classification performance for the sub-database used in the experiment than Gabor wavelet and ridgelet based approaches. The reason is that both transforms are capable of efficiently representing images with 2D singularities. To achieve the same error rate, ripplet-II feature has smaller dimension than ripplet-I.

### 5.2. Image Retrieval

We also conduct experiments to demonstrate the performance of Ripplet-II in content based texture image retrieval.

A sub-database named *Textures* [31] in the texture volume contains 58 images, each of which contains one type of texture. Among the 58 images, 33 images are of size  $512 \times 152$  pixels and 25 images are of size  $1024 \times 1024$  pixels. To test the rotation-invariant capability of different transforms, we need to create rotated versions of the images in the sub-database. To achieve this, we first rotate a texture image by angles from  $0^\circ$  to  $350^\circ$  with a stepsize  $10^\circ$ ; then we crop a patch of size  $128 \times 128$  pixels from the center region of the rotated image. By doing so, we obtain  $58 \times 36 = 2088$  images. Given a image from the sub-database, the algorithm will return a set of images considered

as the rotated version w.r.t. the given image.

Image retrieval is done in the following steps. First, a test image is given as a query to the image retrieval system. Second, apply a feature extraction algorithm (which is the same as that in Section 5.1) to the test image, and obtain a feature vector. Third, apply kNN classifier (where  $k = 35$ ) to the test feature vector and the 2088 images serve as training samples for the kNN classifier; the distance measure used in kNN is  $N_D$ -dimensional Euclidean distance. Assume that in the  $k$  images (which are output of the kNN classifier),  $N_r$  images are rotated versions of the test image. We call  $N_r/k$  as *retrieval rate*, which represents the success rate of image retrieval. We test four types of transforms, i.e., ridgelet, wavelet, ripplelet-II and orthogonal ripplelet-II transform. Ridgelet transform implementation is available [27] and default settings are used. Wavelet transform uses 'Daubechies 4'(db4) wavelets. Ripplelet-II with degree 2 and orthogonal ripplelet-II with degree 2 are used in this experiment. Table 5 and 6 list average retrieval rate using different feature extraction approaches. Table 5 shows that ripplelet-II and orthogonal ripplelet-II transforms have higher average retrieval rate than wavelet transform in most cases. Ripplelet-II and orthogonal ripplelet-II transforms have similar performance in C1, C2 and C3. Results show that C4 is not a good feature for image retrieval. Table 6 shows that ripplelet-II transform achieves higher average retrieval rate than ridgelet and wavelet transform under all feature dimensions tested. Orthogonal ripplelet-II transform outperforms wavelet and ridgelet transform.

Compared to the database in Section 5.1, the database used here has smaller in-class distance. Experimental results show that ripplelet-II works well for both large and small in-class distances. Orthogonal ripplelet-II transform only works for small in-class distance case. The reason is that orthogonal ripplelet-II transform contains no direction information. When there are large rotations involved, features based on orthogonal ripplelet-II transform will not provide similar descriptions about rotated images from the same texture. However, Ripplelet-II transform can capture the directions of content, so ripplelet-II transform outperforms other transforms when applied to texture classification with rotated images.

In Table 7, we present the best performance of various transform based approaches in image retrieval. Gabor wavelet [28, 29] is used for feature extraction. The frequency range is  $[0.05, 0.4]$  and Gabor envelop is calculated accordingly [28, 29]. Gabor wavelet filters with 5 scales and 6 directions are applied to obtain 30 subbands coefficients of an image. The feature vector



Table 5: Average retrieval rate under different transforms using feature extractor 1

Feature dimension $N_D$	C1			
	Wavelet	Ridgelet	Ripplet-II	Orth-Ripplet-II
1	0.4174	0.4521	0.6312	0.8135
4	0.6669	0.6544	0.9948	0.8540
16	0.7034	0.6869	0.9959	0.9409
64	0.7016	0.6632	0.9927	0.9425
256	0.7382	0.6094	0.9876	0.9695
	C2			
1	0.6394	0.4489	0.7653	0.7659
4	0.7182	0.6438	0.7883	0.7692
16	0.7591	0.6079	0.7884	0.8386
64	0.7885	0.5237	0.7894	0.8967
256	0.8070	0.4536	0.7905	0.8876
	C3			
1	0.6384	0.3344	0.7581	0.7536
4	0.8511	0.5968	0.9299	0.7870
16	0.8384	0.5682	0.9413	0.9220
64	0.7863	0.5290	0.9485	0.9632
256	0.7532	0.4923	0.8701	0.7731
	C4			
1	0.1754	0.1810	0.5551	0.2421
4	0.0911	0.1529	0.7660	0.3226
16	0.0539	0.1203	0.7687	0.3963
64	0.0278	0.0883	0.4248	0.3810
256	0.0190	0.0507	0.5305	0.3749

Table 6: Average retrieval rate under different transforms using feature extractor 2

Feature Dimension $N_D$	Ridgelet	Wavelet	Ripplet-II	Orth-Ripplet-II
1	0.5013	0.6426	0.803	0.8016
2	0.5773	0.8513	0.9495	0.8940
4	0.5921	0.8898	0.9831	0.9343
8	0.6343	0.9154	0.986	0.9831
16	0.7054	0.941	0.9869	0.9866

Table 7: Retrieval rate comparisons

Transform	Best retrieval rate
Ridgelet	0.7
Gabor Wavelet	0.47
Ripplet-I	0.91
Ripplet-II	0.98

consists of the mean and standard derivation of each subband as [30]. Then the feature dimension is 60. When we apply ripplet-I [17] to images, we will have ripplet-I coefficients with different scales and different directions. Due to the property of ripplet-I, the numbers of directions at each scale are different. The feature vector of ripplet-I consists of the mean and standard derivation of each subband from the ripplet-I coefficients. In this experiment, the dimension of ripplet-I feature is 38.

Results in Table 7 show that both ripplet-I and ripplet-II achieve better classification performance for the sub-database used in the experiment than Gabor wavelet and ridgelet based approaches. The reason is that both transforms are capable of efficiently representing images with 2D singularities. To achieve the same error rate, ripplet-II feature has smaller dimension than ripplet-I. Results also show that Gabor wavelet is not suitable for small in-class distance case.

## 6. Conclusion

In this paper, we proposed a new transform called ripplelet transform Type II (ripplelet-II) for resolving 2D singularities. Ripplelet-II transform is basically generalized Radon transform followed by 1D wavelet transform. Both forward and inverse ripplelet-II transform were developed for continuous and discrete cases. Ripplelet-II transform with  $d = 2$  can achieve sparser representation for 2D images, compared to ridgelet. Hence, ripplelet-II transform can be used for feature extraction due to its efficiency in representing edges and textures. Ripplelet-II transform also enjoys rotation invariant property, which can be leveraged by applications such as texture classification and image retrieval. Experiments in texture classification and image retrieval demonstrate that the ripplelet-II transform based scheme outperforms wavelet and ridgelet transform based approaches.

## References

- [1] J. Xu, D. Wu, Ripplelet-ii transform for feature extraction, in: Proceedings of SPIE Visual Communications and Image Processing (VCIP), 2010.
- [2] D. L. Donoho, M. Vetterli, R. A. DeVore, I. Daubechies, Data compression and harmonic analysis, IEEE Trans. Inform. Th. 44 (6) (1998) 2435–2476.
- [3] I. Daubechies, Ten Lectures on Wavelets, SIAM, Philadelphia, PA, 1992.
- [4] S. Mallat, A Wavelet Tour of Signal Processing, 2nd Edition, Academic, New York, 1999.
- [5] E. J. Candes, D. L. Donoho, Ridgelets: a key to higherdimensional intermittency?, Phil. Trans. R. Soc. Lond. A. (1999) 2459–2509.
- [6] M. Do, M. Vetterli, The finite ridgelet transform for image representation, IEEE Trans. Image Processing 12 (1) (2003) 16–28.
- [7] S. R. Deans, The Radon Transform and Some of Its Applications, John Wiley & Sons, New York, 1983.
- [8] J. L. Starck, E. J. Candes, D. L. Donoho, The curvelet transform for image denoising, IEEE Trans. Image Processing 11 (2002) 670–684.

- [9] D. L. Donoho, M. R. Duncan, Digital curvelet transform: strategy, implementation and experiments, in: Proc. Aerosense 2000, Wavelet Applications VII. SPIE, Vol. 4056, 2000, pp. 12–29.
- [10] E. Candes, D. Donoho, Continuous curvelet transform: I. Resolution of the wavefront set, Appl. Comput. Harmon. Anal 19 (2) (2005) 162–197.
- [11] E. Candes, D. Donoho, Continuous curvelet transform: II. Discretization and frames, Appl. Comput. Harmon. Anal 19 (2005) 198–222.
- [12] L. Hormander, The Analysis of Linear Partial Differential Operators, Springer-Verlag, Berlin, 2003.
- [13] E. J. Candes, D. L. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise  $c^2$  singularities, Commun. Pure Appl. Math 57 (2) (2003) 219–266.
- [14] M. N. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, IEEE Trans. Image Processing 14 (12) (2005) 2091–2106.
- [15] M. N. Do, M. Vetterli, Contourlets, in: G. V. Welland (Ed.), Beyond wavelets, Academic Press, New York, 2003.
- [16] E. Le Pennec, S. Mallat, Sparse geometric image representations with bandelets, IEEE Trans. Image Processing 14 (4) (2005) 423–438.
- [17] J. Xu, D. Wu, Ripplet transform for feature extraction, in: Proceedings of SPIE Defense Security Symposium, Vol. 6970, 2008, pp. 69700X–69700X–10.
- [18] A. Cormack, The Radon transform on a family of curves in the plane (I), Proceedings of the American Mathematical Society 83 (2) (1981) 325–330.
- [19] A. Cormack, The Radon transform on a family of curves in the plane (II), Proceedings of the American Mathematical Society 86 (2) (1982) 293–298.
- [20] F. Natterer, The mathematics of computerized tomography, SIAM, 2001.

- [21] R. Bracewell, The fourier transform, *Scientific American* 260 (6) (1989) 86–95.
- [22] K. Denecker, J. Van Overloop, F. Sommen, The general quadratic Radon transform, *Inverse Problems* 14 (3) (1998) 615–634.
- [23] Usc-sipi image database, <http://sipi.usc.edu/database>.
- [24] Rotated textures, <http://sipi.usc.edu/database/database.cgi?volume=rotate>.
- [25] C. Pun, M. Lee, Log-polar wavelet energy signatures for rotation and scale invariant texture classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (5) (2003) 590–603.
- [26] I. Jolliffe, *Principal component analysis*, Springer verlag, 2002.
- [27] Frit toolbox, <http://www.ifp.illinois.edu/minhdo/software>.
- [28] B. S. Manjunath, W. Ma, Texture features for browsing and retrieval of image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI - Special issue on Digital Libraries)* 18 (8) (1996) 837–42.
- [29] Gabor wavelet, <http://vision.ece.ucsb.edu/texture/software/>.
- [30] S. Arivazhagan, L. Ganesan, S. Priyal, Texture classification using Gabor wavelets based rotation invariant features, *Pattern Recognition Letters* 27 (16) (2006) 1976–1982.
- [31] Textures, <http://sipi.usc.edu/database/database.cgi?volume=textures>.