Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 592928, 15 pages
http://dx.doi.org/10.1155/2014/592928

Research Article

Fingerprint Classification Combining Curvelet Transform and Gray-Level Cooccurrence Matrix

Jing Luo,[1,2,3] Dan Song,[1,2] Chunbo Xiu,[1,2] Shuze Geng,[1,2] and Tingting Dong[1,2]

[1] *Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tianjin 300387, China*
[2] *College of Electrical Engineering and Automation, Tianjin Polytechnic University, Tianjin 300387, China*
[3] *School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia*

Correspondence should be addressed to Dan Song; sweet0502@sina.com

Received 30 April 2014; Revised 25 June 2014; Accepted 1 July 2014; Published 17 August 2014

Academic Editor: L. W. Zhang

Copyright © 2014 Jing Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fingerprint classification is an important indexing scheme to reduce fingerprint matching time for a large database for efficient large-scale identification. The abilities of Curvelet transform capturing directional edges of fingerprint images make the fingerprint suitable to be classified for higher classification accuracy. This paper presents an efficient algorithm for fingerprint classification combining Curvelet transform (CT) and gray-level cooccurrence matrix (GLCM). Firstly, we use fast discrete Curvelet transform warping (FDCT_WARPING) to decompose the original image into five scales Curvelet coefficients and construct the Curvelet filter by Curvelet coefficients relationship at adjacent scales to remove the noise from signals. Secondly, we compute the GLCMs of Curvelet coefficients at the coarsest scale and calculate 16 texture features based on 4 GLCMs. Thirdly, we construct 49 direction features of Curvelet coefficients at the other four scales. Finally, fingerprint classification is accomplished by K-nearest neighbor classifiers. Extensive experiments were performed on 4000 images in the NIST-4 database. The proposed algorithm achieves the classification accuracy of 94.6 percent for the five-class classification problem and 96.8 percent for the four-class classification problem with 1.8 percent rejection, respectively. The experimental results verify that proposed algorithm has higher recognition rate than that of wavelet-based techniques.

1. Introduction

As a type of human biometrics, fingerprint has been widely used for personal recognition in forensic and civilian applications because of its uniqueness, immutability, and low cost. An automatic recognition of people based on fingerprints requires matching of an input fingerprint with a large number of fingerprints in a database. However, the database can be huge (e.g., the FBI database contains more than 70 million fingerprints), such a task can be very expensive in terms of computational burden and time. In order to reduce the search time and computational burden, fingerprints in the database are classified into several prespecified types or subclasses. When an input fingerprint is received, a coarse level matching is applied to determine which subclass the input belongs to, and then at a finer level, it is compared to samples within the subset of the database for recognition. While such a scheme is obviously more efficient, the first step, that is, fingerprint classification, must be accurate and reliable and hence has attracted extensive research in recent years [1–13].

Fingerprints are classified based on their shapes, and in literature it is common to have five classes as shown in Figure 1 (x):(A) (x = a, b, c, d, e), including whorl (W), right loop (R), left loop (L), arch (A), and tent arch (T). Although these five classes appear very different to us as a person, automatically classifying a fingerprint by a machine is in fact a very challenging pattern recognition problem, due to the small interclass variability, the large intraclass variability and the difficulty for poor quality fingerprints. Fingerprint classification is carried out by analysis and comparison of the features. Over the past decade various approaches have been proposed by means of different types of features, such as singularities [1–3], orientation field [4–6], and statistical and frequency features [7–10]. The methods based on singularities [1–3] accomplish fingerprint classification according to the number and relative position of the core and delta points.

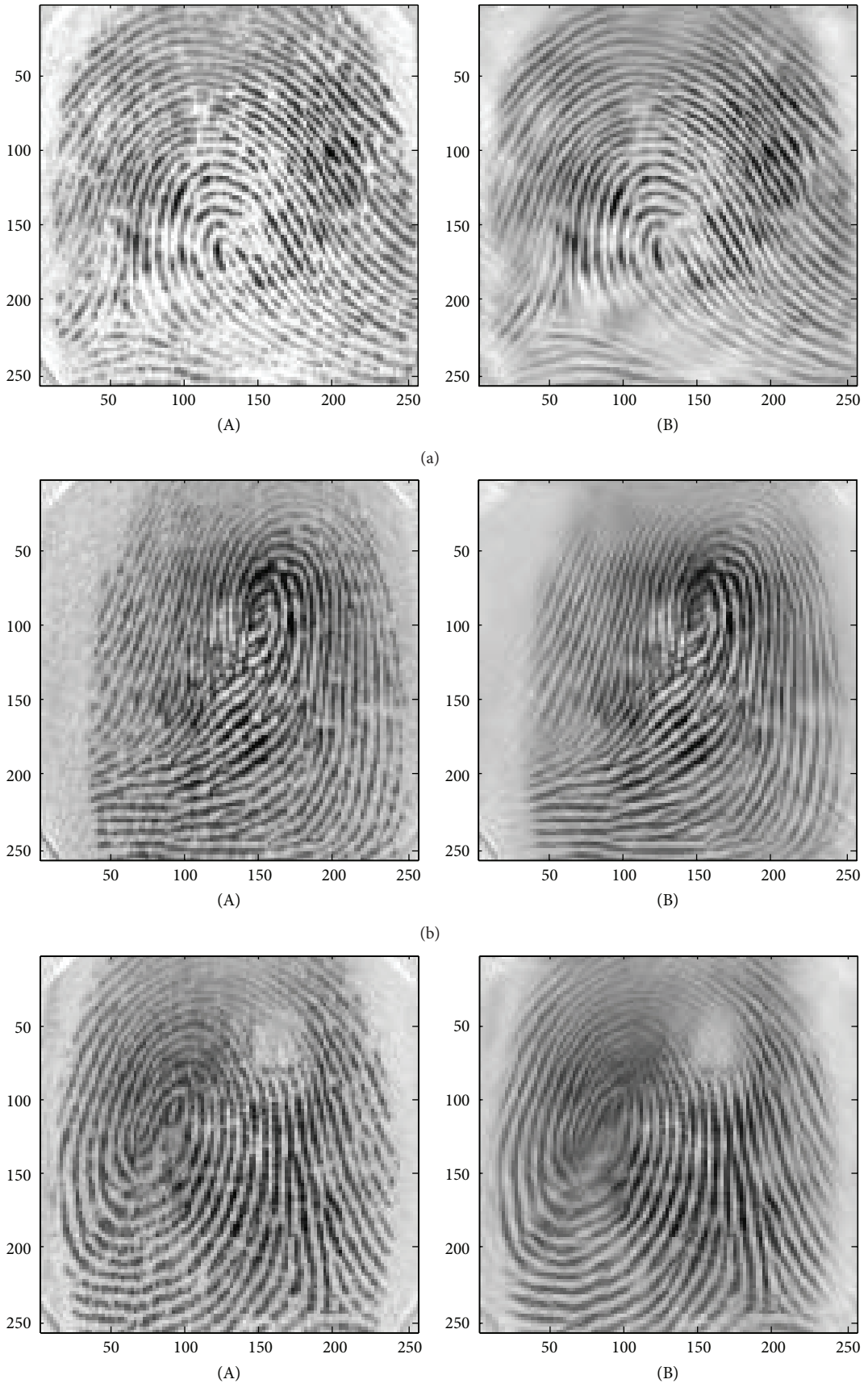


Figure 1: Continued.

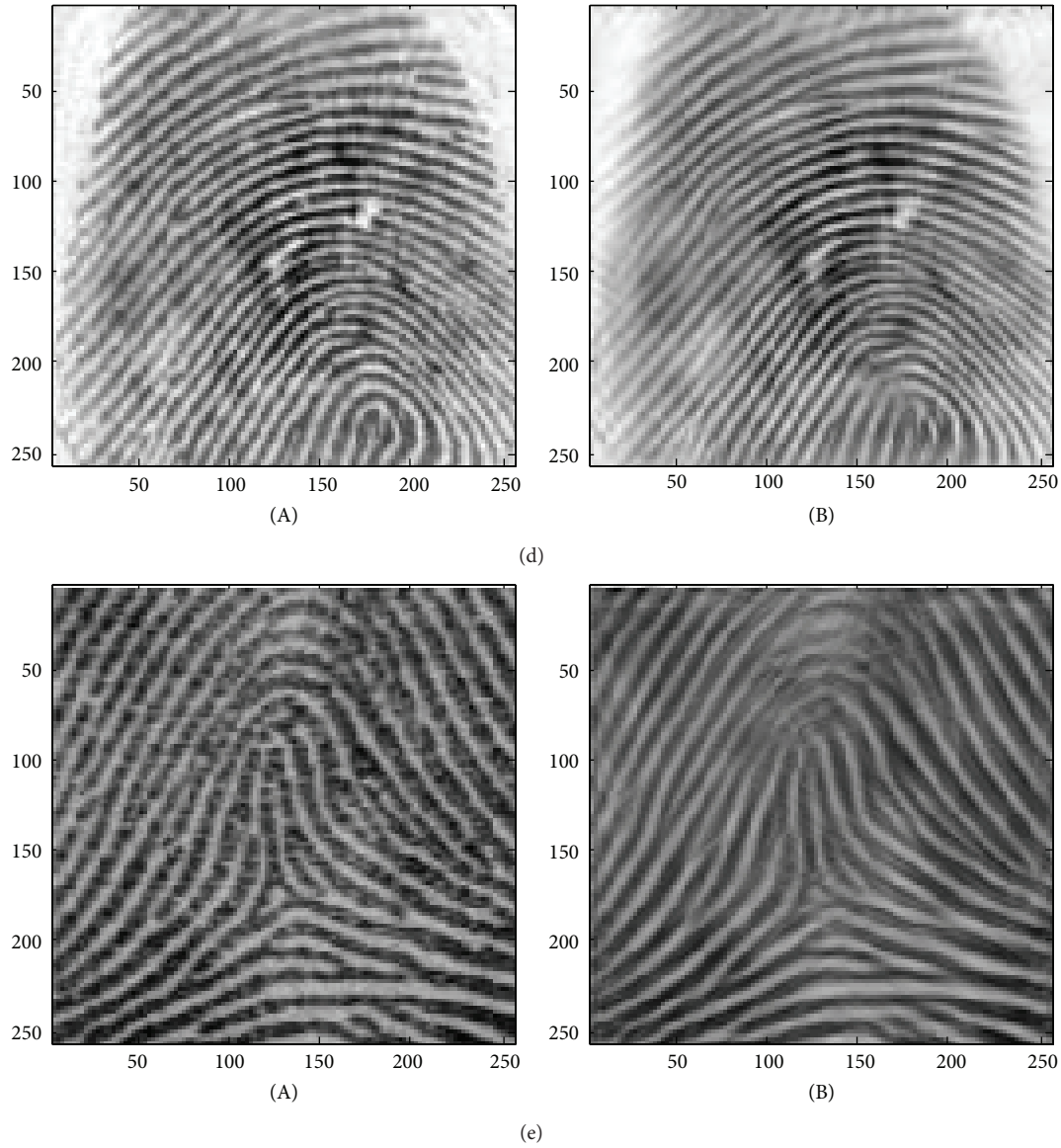


Figure 1: (a) (A) Original right loop image and (B) denoised right loop image. (b) (A) Original left loop image, (B) denoised left loop image. (c) (A) Original whorl image and (B) denoised whorl image. (d) (A) Original arch image and (B) denoised arch image. (e) (A) Original tented arch image and (B) denoised tented arch image.

The approaches using orientation field [4–6] partition the fingerprint orientation field into "homogeneous" orientation regions and the relational graphs of these regions are used to make fingerprint classification. Gabor filter [9] can also be used to extract the fingerprint features for classification where the input image is decomposed into four component images by four Gabor filters, and the standard deviation of the component images in each sector to generate the feature vector, and uses a K-nearest neighbor classifier to make fingerprint classification.

In 2001, Tico et al. [10] proposed an approach to use wavelet transform as features for fingerprint recognition. In [10], a wavelet decomposition on 6 octaves of each fingerprint image was performed, and the normalized l_2-norm of each wavelet subimage is computed to extract a feature vector of length 18. The experimental database contains 168 fingerprint images collected from 21 fingers (8 images per finger). The algorithm achieves the accuracy of 100% when wavelet basis Symmlet 6 or Symmlet 9 was employed. The work in [10] shows that the wavelet features are suitable for matching complex patterns of oriented texture such as fingerprints. However, wavelets are characterized by isotropic scaling (e.g., the standard orthogonal wavelet transform contains wavelets in the directions of primary vertical, primary horizontal, and primary diagonal only) and hence their abilities to resolve directional features are limited. So wavelets are not able to detect curved singularities effectively.

Inspired by the success of wavelets, a number of other multiresolution analysis tools have also been proposed with the aim to present better the edges and other singularities

along curves. These tools include contourlet, ridgelet, and Curvelet. In recent years, researchers have used Curvelets for fingerprint image enhancement [11, 12] and for fingerprint recognition [13]. Comparing to the limited directional features of wavelets, Curvelets are more powerful as they are able to describe a signal by a group of matrices at multiscale and multidirection. Furthermore, with the increase of the scale, the number of directions is much finer.

In 2008, Mandal and Wu [13] proposed to use Curvelet transform for fingerprint recognition and achieved the accuracy of 100%. However, the performance of the proposed algorithm was only tested on a small database of 120 fingerprint images (containing only 15 individuals). Furthermore, in order to ensure the accuracy, the technique requires manual detection of the core of the fingerprint image. Also, before extracting the Curvelet features, the technique needs complex image enhancement process which includes estimation of local ridge-orientation, estimation of local ridge-frequency across the fingerprint, filtering of the image, and binarization (conversion of a gray-scale fingerprint image to binary image).

In this paper, we present a novel fingerprint classification algorithm. Firstly, we use fast discrete Curvelet transform warping (FDCT_WARPING) to decompose the original image into five-scale Curvelet coefficients and construct the Curvelet filter by Curvelet coefficients relationship at adjacent scales to smooth the discontinuities of ridges and remove the noise in the fingerprint image. Secondly, we calculate four gray-level cooccurrence matrices (GLCMS) of Curvelet coefficients at the coarsest scale and calculate 16 texture features based on 4 GLCMS. Furthermore, we construct 49 direction features of Curvelet coefficients at the other four scales. Finally, these combined Curvelet-GLCM based features act as the feature set to a K-nearest neighbor classifier.

In the following sections, we will present the details of our fingerprint classification algorithm. Section 2 presents our noise filtration scheme and our feature extraction scheme. Section 3 presents some experimental results achieved on NIST-4 databases. Finally Section 4 draws the conclusions and outlines the open problems.

2. Fingerprint Classification

2.1. Fingerprint Alignments. Considering the translation and rotation between template images and probe images, this paper adopted the algorithm in [14] to accomplish the fingerprint image registration. The algorithm used the reference points of the central area to calculate the parameters of translation and rotation, which is much more general when there are no cores in the fingerprint images.

2.2. Fast Discrete Curvelet Transform (FDCT). Curvelets were proposed by Candese and Donoho [15], constituting a family of frames that are designed to represent the edges and other singularities along curves. Conceptually, the Curvelet transform is a multiscale pyramid with many orientations and positions at each length scale and needle-shaped elements at fine scale. This pyramid is nonstandard, however. Indeed, Curvelets have useful geometric features that set them apart from wavelets and the likes. For instance, Curvelets present highly anisotropic behavior as it has both variable *length* and *width*. At fine scale, anisotropy increases with decreasing scale, in keeping with power law.

In 2006, Candès et al. proposed two fast discrete Curvelet transforms (FDCT) [16]. The first one is based on unequally spaced fast Fourier transforms (USFFT) [16], and the other is based on the wrapping of specially selected Fourier samples (FDCT_WARPING) [16]. Curvelets by warping have been used for this work, because this is the fastest Curvelet transform currently available [16].

After Curvelet transform, several groups of Curvelet coefficients are generated at different scales and angles. Curvelet coefficients at scale j and angle l are represented by a matrix $C_{j,l}$, and scale j is from finest to coarsest scale, and angle l starts at the top-left corner and increases clockwise.

Suppose that $f(t_1, t_2)$, $1 \le t_1 \le N_1$, $1 \le t_2 \le N_2$ denotes original image and $\widehat{f}[n_1, n_2]$ denotes 2D discrete Fourier transform; N_1, N_2 is the size of original image.

The implementation of FDCT_WARPING is as follows.

Step 1. 2D FFT (fast Fourier transform) is applied on $f(t_1, t_2)$ to obtain Fourier samples $\widehat{f}[n_1, n_2]$.

Step 2. Resample $\widehat{f}[n_1, n_2]$ at each pair of scale and direction j, l in frequency domain, yielding the new sampling function:

$$\widehat{f}\left[n_1, n_2 - n_1 \tan\theta_l\right], \quad (n_1, n_2) \in P_j, \tag{1}$$

where $P_j = \{(n_1, n_2), n_{1,0} \le n_1 < n_{1,0} + L_{1,j}, n_{2,0} \le n_2 < n_{2,0} + L_{2,j}\}$ and $n_{1,0}$ and $n_{2,0}$ are two initial positions of the window function $\widetilde{u}_{j,l}[n_1, n_2]$.

$L_{1,j}$ and $L_{2,j}$ are relevant parameters of 2^j and $2^{j/2}$, respectively, and they are length and width components of window function support interval.

Step 3. Multiplication of the new sampling function $\widehat{f}[n_1, n_2 - n_1 \tan\theta_l]$ with window function $\widetilde{u}_j[n_1, n_2]$, and the result is [16]

$$\widetilde{f}_{j,l}\left[n_1, n_2\right] = \widehat{f}\left[n_1, n_2 - n_1 \tan\theta_l\right] \widetilde{u}_{j,l}\left[n_1, n_2\right], \tag{2}$$

where

$$\begin{aligned}
\widetilde{u}_{j,l}\left[n_1, n_2\right] &= w_j\left(w_1, w_2\right) v_j\left(s_{\theta_l} \cdot \frac{\left(2^{\lfloor j/2 \rfloor} w_2\right)}{w_1}\right), \\
w_j\left(w_1, w_2\right) &= \sqrt{\phi_{j+1}^2\left(w^2\right) - \phi_j^2\left(w^2\right)}, \\
\phi_j\left(w_1, w_2\right) &= \phi\left(2^{-j} w_1\right) \phi\left(2^{-j} w_2\right), \\
s_{\theta_l} &= \begin{bmatrix} 1 & 0 \\ -\tan\theta_l & 1 \end{bmatrix}, \\
\tan\theta_l &= l \times 2^{\lfloor -j/2 \rfloor}, \quad l = -2^{\lfloor -j/2 \rfloor}, \ldots, 2^{\lfloor -j/2 \rfloor} - 1.
\end{aligned} \tag{3}$$

Step 4. Apply the inverse 2D FFT to each $\widetilde{f}_{j,l}$, hence collecting the discrete coefficients $C_{j,l}$.

2.3. Fingerprint Image FDCT and Noise Filtration Technique. Curvelet transform describes a signal by the power at each scale, angle, and position. Curvelets have variable length in addition to variable width. Curvelet transform has improved directional capability and better ability to represent edges and other singularities along curves compared to other multiscale transforms, for example, wavelet transform. Image like fingerprints often have two nearby regions (ridges and valleys) that differ in pixel values. These variations in pixel values between two consecutive regions are likely to form such "edges" and this edge information is eventually captured by digital Curvelet transform.

2.3.1. Acquisition of Curvelet Coefficients. In this paper, fingerprint images are from NIST-4 databases. Firstly, the size of all the images is normalized to 256 × 256 pixels. The proposed algorithm is implemented by using MATLAB 7.5 programming language and is executed under Windows XP Professional O.S. on a PC with AMD Dual-Core CPU E-350 at 1.28 GHz. The original image is decomposed by FDCT_WARPING [16].

Suppose that $f(t_1, t_2)$, $1 \le t_1 \le N_1$, $1 \le t_2 \le N_2$ denotes original fingerprint image and $\widehat{f}[n_1, n_2]$ denotes 2D discrete Fourier transform; N_1, N_2 is the size of original fingerprint image.

The number of scales can be calculated by

$$N_{\text{scales}} = \text{ceil}\left(\log_2\left(\min\left(N_1, N_2\right)\right) - 3\right), \tag{4}$$

where $\min(N_1, N_2)$ returns the minimum of N_1 and N_2 and ceil(·) rounds the input to the nearest integers greater than or equal to input.

The number of orientation at scale $2, 3, \ldots, N_{\text{scales}} - 1$, is calculated by

$$l_j = 16 \times 2^{\text{ceil}((N_{\text{scales}} - i)/2)}, \tag{5}$$

where $i = N_{\text{scales}}$, $j = 2$; $i = N_{\text{scales}} - 1$, $j = 3; \ldots; i = 3$, and $j = N_{\text{scales}} - 1$.

Noted that if FDCT_WARPING chooses the Curvelets for the coefficients at the finest level N_{scales}, the number of orientation at scale N_{scales} can be determined by (2). On the other hand, when choosing the wavelets for the coefficients at the finest level N_{scales}, there is only one angle at the finest level N_{scales}. In this paper, we adopt the wavelets for the coefficients at the finest level N_{scales}. As a result, there is only one angle at the finest level N_{scales}.

For scale $j = 2, 3, \ldots, (N_{\text{scales}} - 1)$, all the Curvelet coefficients are divided into the four quadrants. The quadrant label of Curvelet coefficients is denoted by quad, and four quadrants are denoted by quad $= 1, 2, 3, 4$, respectively. At each quadrant, Curvelet coefficients are further subdivided in angular panels. The number of angular panels at each quadrant of each scale j, denoted by $n_{\text{quad}-j}$, can be calculated as follows:

$$n_{\text{quad}-j} = 4 \times 2^{\text{ceil}((N_{\text{scales}} - i)/2)}, \tag{6}$$

where $i = N_{\text{scales}}$, $j = 2$; $i = N_{\text{scales}} - 1$, $j = 3; \ldots; i = 3$, and $j = N_{\text{scales}} - 1$.

In this paper, according to (4), we can get the number of scales $N_{\text{scales}} = 5$. According to (5), the number of angle at scale 4 is 32, at scale 3 is 32, and at scale 2 is 16. At scale 4, the number of angular panels of each quadrant is 8. At scale 3, the number of angular panels of each quadrant is 8. At scale 2, the number of angular panels of each quadrant is 4. After decomposition, the original image was divided into three levels: coarse, detail, and fine. The low-frequency coefficients were assigned to coarse. The high-frequency coefficients were assigned to fine. The middle-frequency coefficients were assigned to detail. According to FDCT_WARPING [16], the scale j is from finest to coarsest scale and angle l starts at the top-left corner and increases clockwise.

The acquisition of Curvelet coefficients is as follows.

Step 1. 2D FFT (fast Fourier transform) is applied on $f(t_1, t_2)$ to obtain Fourier samples $\widehat{f}[n_1, n_2]$.

Step 2. Acquire the Curvelet coefficients at scale 5, denoted by matrix $C_{5,1}$.

Suppose that

$$\begin{aligned} M_1 &= \frac{N_1}{6}, \qquad M_2 = \frac{N_2}{6}, \\ l_{N_{\text{scales}}-1} &= \text{floor}\left(2 \times M_1\right) - \text{floor}\left(M_1\right) - 1, \\ l_{N_{\text{scales}}-2} &= \text{floor}\left(2 \times M_2\right) - \text{floor}\left(M_2\right) - 1. \end{aligned} \tag{7}$$

In this paper, floor(·) rounds the input to the nearest integers less than or equal to the input.

(1) Construct the right and left windows along the horizontal direction, denoted by row vector W_{r-1} and W_{l-1}, respectively. Consider

$$W_{r-1}(i) = \begin{cases} 1, & \\ & i = 0, \\ e^{\left(1 - 1/\left(1 - e^{(1-1/i)}\right)\right)}, & \\ & i = \dfrac{1}{l_{N_{\text{scales}}-1}}, \dfrac{2}{l_{N_{\text{scales}}-1}}, \ldots, \dfrac{l_{N_{\text{scales}}-1} - 1}{l_{N_{\text{scales}}-1}}, \\ 0, & \\ & i = 1, \end{cases}$$

$$W_{l-1}(i) = \begin{cases} 0, & \\ & i = 0, \\ e^{\left(1 - 1/\left(1 - e^{(1-1/(1-i))}\right)\right)}, & \\ & i = \dfrac{1}{l_{N_{\text{scales}}-1}}, \dfrac{2}{l_{N_{\text{scales}}-1}}, \ldots, \dfrac{l_{N_{\text{scales}}-1} - 1}{l_{N_{\text{scales}}-1}}, \\ 1, & \\ & i = 1. \end{cases} \tag{8}$$

(2) Construct the right and left windows along the vertical direction, denoted by row vector W_{r-2} and W_{l-2}, respectively. Consider

$$W_{r-2}(i) = \begin{cases} 1, & i = 0, \\ e^{(1-1/(1-e^{(1-1/i)}))}, & i = \dfrac{1}{l_{N_{\text{scales}}-2}}, \dfrac{2}{l_{N_{\text{scales}}-2}}, \ldots, \dfrac{l_{N_{\text{scales}}-2}-1}{l_{N_{\text{scales}}-2}}, \\ 0, & i = 1, \end{cases} \tag{9}$$

$$W_{l-2}(i) = \begin{cases} 0, & i = 0, \\ e^{(1-1/(1-e^{(1-1/(1-i))}))}, & i = \dfrac{1}{l_{N_{\text{scales}}-2}}, \dfrac{2}{l_{N_{\text{scales}}-2}}, \ldots, \dfrac{l_{N_{\text{scales}}-2}-1}{l_{N_{\text{scales}}-2}}, \\ 1, & i = 1. \end{cases} \tag{10}$$

W_{r-1}, W_{l-1}, W_{r-2}, and W_{l-2} are normalized by

$$\begin{aligned} W_{r-1}(i) &= \frac{W_{r-1}(i)}{\sqrt{[W_{r-1}(i)]^2 + [W_{l-1}(i)]^2}}, \\ W_{l-1}(i) &= \frac{W_{l-1}(i)}{\sqrt{[W_{r-1}(i)]^2 + [W_{l-1}(i)]^2}}, \\ W_{r-2}(i) &= \frac{W_{r-2}(i)}{\sqrt{[W_{r-2}(i)]^2 + [W_{l-2}(i)]^2}}, \\ W_{l-2}(i) &= \frac{W_{l-2}(i)}{\sqrt{[W_{r-2}(i)]^2 + [W_{l-2}(i)]^2}}. \end{aligned} \tag{11}$$

(3) Construct the two sub-low-pass filters, denoted by row vector $f_{\text{lowpasssub1}}$ and $f_{\text{lowpasssub2}}$, respectively. Consider

$$\begin{aligned} f_{\text{lowpasssub1}} &= \left[W_{l-1}, \underbrace{1, 1 \cdots 1}_{p_1}, W_{r-1} \right]_{1\times[2\times\text{floor}(2\times M_1)+1]}, \\ f_{\text{lowpasssub2}} &= \left[W_{l-2}, \underbrace{1, 1 \cdots 1}_{p_2}, W_{r-2} \right]_{1\times[2\times\text{floor}(2\times M_2)+1]}, \end{aligned} \tag{12}$$

where $p_1 = 2 \times \text{floor}(M_1) + 1$, $p_2 = 2 \times \text{floor}(M_2) + 1$.

(4) Construct a low-pass filter at scale 5, denoted by matrix $(f_{\text{lowpass_5}})_{[2\times\text{floor}(2\times M_1)+1]\times[2\times\text{floor}(2\times M_2)+1]}$, and

$$f_{\text{lowpass_5}} = \left(f_{\text{lowpasssub1}}\right)^T \times f_{\text{lowpasssub2}}, \tag{13}$$

where $(\cdot)^T$ is the transpose vector or matrix of the input vector or matrix.

(5) Construct a high-pass filter at scale 5, denoted by matrix $f_{\text{hipass_5}}$, which has the same size as $f_{\text{lowpass_5}}$. Consider

$$f_{\text{hipass_5}} = \sqrt{1 - \left(f_{\text{lowpass_5}}\right)^2}. \tag{14}$$

(6) $\widehat{f}[n_1, n_2]$ is filtered by $f_{\text{hipass_5}}$, hence generating the filtered high-pass signal at scale 5 $f'_{\text{hipass_5}}$, which has the same size of $N_1 \times N_2$ as $\widehat{f}[n_1, n_2]$. Consider

$$f'_{\text{hipass_5}}(n_1, n_2) = \begin{cases} \widehat{f}(n_1, n_2) \times f_{\text{hipass_5}}(n_1, n_2), & a \le n_1 \le b,\ c \le n_2 \le d, \\ \widehat{f}(n_1, n_2), & \text{others}, \end{cases} \tag{15}$$

where the filter at scale 5 has the following range:

$$\begin{aligned} a &= -\text{floor}(2 \times M_1) + \text{ceil}\left(\frac{(N_1+1)}{2}\right), \\ b &= \text{floor}(2 \times M_1) + \text{ceil}\left(\frac{(N_1+1)}{2}\right), \\ c &= -\text{floor}(2 \times M_2) + \text{ceil}\left(\frac{(N_2+1)}{2}\right), \\ d &= -\text{floor}(2 \times M_2) + \text{ceil}\left(\frac{(N_2+1)}{2}\right). \end{aligned} \tag{16}$$

(7) Inverse 2DFFT (inverse fast Fourier transform) is applied to $f'_{\text{hipass_5}}$, hence generating the discrete Curvelet coefficients at scale 5, $C_{5,1}$.

(8) $\widehat{f}[n_1, n_2]$ is filtered by $f_{\text{lowpass_5}}$, hence generating the filtered low-pass signal at scale 5, $(f'_{\text{lowpass_5}})_{[2\times\text{floor}(2\times M_1)+1]\times[2\times\text{floor}(2\times M_2)+1]}$ and

$$\begin{aligned} f'_{\text{lowpass}_5}(n_1, n_2) &= \widehat{f}(n_1, n_2) \times f_{\text{lowpass}_5}(n_1, n_2), \\ & a \le n_1 \le b, \quad c \le n_2 \le d. \end{aligned} \tag{17}$$

Step 3. Acquire the Curvelet coefficients at scale 4 and angle 1 to angle 32, $C_{4,l}$, $l = 1, 2, \ldots, 32$.

Firstly, we acquire the Curvelet coefficients at scale 4 and angle 1, $C_{4,1}$.

Suppose that

$$M_1 = \frac{N_1}{(6 \times 2)}, \quad M_2 = \frac{N_2}{(6 \times 2)}. \tag{18}$$

The filter at scale 4 has the following range:

$$\begin{aligned} a_{N_{\text{scales}}-1} &= -\text{floor}(2 \times M_1) + \text{floor}(4 \times M_1) + 1, \\ b_{N_{\text{scales}}-1} &= \text{floor}(2 \times M_1) + \text{floor}(4 \times M_1) + 1, \\ c_{N_{\text{scales}}-1} &= -\text{floor}(2 \times M_2) + \text{floor}(4 \times M_2) + 1, \\ d_{N_{\text{scales}}-1} &= \text{floor}(2 \times M_2) + \text{floor}(4 \times M_2) + 1. \end{aligned} \tag{19}$$

(1) Construct a low-pass filter at scale 4 and angle 1 in the same way as at scale 5, $(f_{\text{lowpass_4}})_{[2\times\text{floor}(2\times M_1)+1]\times[2\times\text{floor}(2\times M_2)+1]}$.

(2) Construct a high-pass filter at scale 4 and angle 1 in the same way as at scale 5, $(f_{\text{hipass_4}})_{[2\times\text{floor}(2\times M_1)+1]\times[2\times\text{floor}(2\times M_2)+1]}$.

(3) $f'_{\text{lowpass_5}}$ is filtered by $f_{\text{lowpass_4}}$, hence generating the filtered low-pass signal at scale 4 $(f'_{\text{lowpass_4}})_{[2\times\text{floor}(2\times M_1)+1]\times[2\times\text{floor}(2\times M_2)+1]}$.

(4) $f'_{\text{lowpass_5}}$ is filtered by $f_{\text{hipass_4}}$, hence generating the filtered high-pass signal at scale 4 $f'_{\text{hipass_4}}$, which has the same size as that of $f'_{\text{lowpass_5}}$.

(5) Determine the discrete locating window of wedge wave at scale 4 and angle 1.

The Curvelet coefficients at scale 4 are divided into the four quadrants. The quadrant label of Curvelet coefficients is denoted by quad, and four quadrants are denoted by quad = 1, 2, 3, 4, respectively. Each quadrant has 8 angles. In the first quadrant, quad = 1, angle ranges from 1 to 8, in the second quadrants, quad = 2, angle ranges from 9 to 16, in the third quadrants, quad = 3, angle ranges from 17 to 24, in the fourth quadrants, and quad = 4, angle ranges from 25 to 32.

Suppose that

$$\begin{aligned} &M_{\text{horiz}} = M_2, \\ &M_{\text{vert}} = M_1, \end{aligned} \quad \text{quad} = 1 \text{ or } 3,$$

$$\begin{aligned} &M_{\text{horiz}} = M_1, \\ &M_{\text{vert}} = M_2, \end{aligned} \quad \text{quad} = 2 \text{ or } 4, \tag{20}$$

$$p = \frac{1}{2 \times n_{\text{quad}-4}},$$

$$q = 2 \times \text{floor}\left(4 \times M_{\text{horiz}}\right),$$

where $n_{\text{quad}-4}$ denotes the number of angle at each quadrant at scale 4; in this paper, $n_{\text{quad}-4} = 8$.

The left vector of wedge wave is denoted by $W_{\text{left}} = (a_{1,k+1})_{1\times(n_{\text{quad}-4}+1)}$, where $a_{1,k+1} = \text{round}\ (k \times p \times q + 1)$, $k = 0, 1, 2, \ldots, n_{\text{quad}-4}$.

The right vector of wedge wave is denoted by $W_{\text{right}} = (a_{1,k+1})_{1\times(n_{\text{quad}-4})}$, where $a_{1,k+1} = q + 2 - \text{round}\ (k \times p \times q + 1)$, $k = n_{\text{quad}-4}, n_{\text{quad}-4} - 1, \ldots, 1$.

The combination wedge wave vector is denoted by $W = [W_{\text{left}}, W_{\text{right}}]_{1\times(2\times n_{\text{quad}-4}+1)}$.

The endpoint vector of wedge wave is denoted by $W_{\text{end}} = (a_{1,k})_{1\times(n_{\text{quad}-4})}$, where $a_{1,k} = W(2 \times k) \cdots k = 1, 2, \ldots, n_{\text{quad}-4}$.

The first midpoint vector of wedge wave is denoted by $W_{\text{mid1}} = (a_{1,k})_{1\times(n_{\text{quad}-4}-1)}$, where $a_{1,k} = W_{\text{end}}(k) \cdots k = 1, 2, \ldots, n_{\text{quad}-4} - 1$.

The second midpoint vector of wedge wave is denoted by $W_{\text{mid2}} = (a_{1,k})_{1\times(n_{\text{quad}-4}-1)}$, where $a_{1,k} = W_{\text{end}}(k) \cdots k = 2, 3, \ldots, n_{\text{quad}-4}$.

The combination midpoint vector of wedge wave is denoted by $W_{\text{mid}} = (a_{1,k})_{1\times(n_{\text{quad}-4}-1)}$, where $a_{1,k} = ((W_{\text{mid1}}(k) + W_{\text{mid2}}(k))/2) \cdots k = 1, 2, 3, \ldots, n_{\text{quad}-4} - 1$.

The first wedge wave endpoint along the vertical orientation is denoted by

$$r_{\text{first_wedge_endpont}} = \text{round}\left(\frac{2 \times \text{floor}\left(4 \times M_{\text{vert}}\right)}{2 \times n_{\text{quad}-4}} + 1\right). \tag{21}$$

The length of the first wedge wave is denoted by

$$\begin{aligned} s_{\text{wedge}} = {} & \text{floor}\left(4 \times M_{\text{vert}}\right) - \text{floor}\left(M_{\text{vert}}\right) \\ & + \text{ceil}\left(\frac{r_{\text{first_wedge_endpont}}}{4}\right). \end{aligned} \tag{22}$$

The width of the wedge wave is denoted by $W_{\text{wedge}} = W_{\text{end}}(2) + W_{\text{end}}(1) - 1$.

The slope of the first wedge wave is $s_{\text{slope}} = [\text{floor}(4 \times M_{\text{horiz}}) + 1 - W_{\text{end}}(1)]/\text{floor}(4 \times M_{\text{vert}})$.

The left line vector is denoted by $s_{\text{left_line}} = (a_{1,k})_{1\times(s_{\text{wedge}})}$, where $a_{1,k} = \text{round}(2 - W_{\text{end}}(1) + s_{\text{slope}} \times (k - 1)) \cdots k = 1, 2, \ldots, s_{\text{wedge}}$.

The first row coordinate is $r_{\text{first}} = \text{floor}(4 \times M_{\text{vert}}) + 2 - \text{ceil}((s_{\text{wedge}} + 1)/2)$.

The first column coordinate is $c_{\text{first}} = \text{floor}(4 \times M_{\text{horiz}}) + 2 - \text{ceil}((W_{\text{wedge}} + 1)/2)$.

Condition column vector is denoted by $V_{\text{col}} = (a_{1,k})_{1\times W_{\text{wedge}}}$, where

$$\begin{aligned} a_{1,k} = {} & s_{\text{left_line}}\,(k+1) \\ & + \text{mod}\left(\left(k - s_{\text{left_line}}\,(k+1) + c_{\text{first}}\right), W_{\text{wedge}}\right), \\ & k = 0, 1, 2, \ldots, W_{\text{wedge}} - 1, \end{aligned} \tag{23}$$

$$\text{mod}\,(x, y) = \begin{cases} x - n \times y \quad n = \text{floor}\left(\dfrac{x}{y}\right), & y \neq 0, \\ x, & y = 0. \end{cases}$$

Thus, the discrete locating window of wedge wave is denoted by $W_{\text{data}} = (a_{n_1,n_2})_{s_{\text{wedge}}\times W_{\text{wedge}}}$, where

$$a_{n_1,n_2} = \begin{cases} f'_{\text{hipass}_4}\,(n_1, n_2), & V_{\text{col}}\,(n_2) > 0, \\ 0, & V_{\text{col}}\,(n_2) \leq 0, \end{cases} \tag{24}$$

where

$$n_1 = 1 + \text{mod}\left(k - r_{\text{first}}, s_{\text{wedge}}\right), \quad k = 1, 2, \ldots, s_{\text{wedge}}, \tag{25}$$

$$\begin{aligned} n_2 = {} & 0.5 \times V_{\text{col}}\,(k) + 1 + \left|V_{\text{col}}\,(k) - 1\right|, \\ & k = 1, 2, \ldots, W_{\text{wedge}}. \end{aligned} \tag{26}$$

(6) The discrete locating window of wedge wave W_{data} is filtered and rotated, hence generating matrix $W_{\text{data_2}}$.

Suppose that

$$y_{\text{corner}} = \left(a_{1,k}\right)_{1\times s_{\text{wedge}}}, \tag{27}$$

where $a_{1,k} = k, k = 1, 2, 3, \ldots, s_{\text{wedge}}$ and

$$x_{\text{corner}} = (a_{1,k})_{1\times q}, \tag{28}$$

where $a_{1,k} = k, k = 1, 2, 3, \ldots, q+1$ and

$$X = \begin{bmatrix} x_{\text{corner}} \\ x_{\text{corner}} \\ x_{\text{corner}} \\ \vdots \\ x_{\text{corner}} \end{bmatrix}_{s_{\text{wedge}}\times q},$$

$$Y^T = \begin{bmatrix} y_{\text{corner}} \\ y_{\text{corner}} \\ y_{\text{corner}} \\ \vdots \\ y_{\text{corner}} \end{bmatrix}_{q\times s_{\text{wedge}}}, \tag{29}$$

$$X_{\text{wrap}} = \left(X\left(n_1, n_2\right)\right)_{s_{\text{wedge}}\times W_{\text{wedge}}},$$

where n_1, n_2 are calculated by (25) and (26), respectively. Consider

$$Y_{\text{wrap}} = \left(Y\left(n_1, n_2\right)\right)_{s_{\text{wedge}}\times W_{\text{wedge}}}, \tag{30}$$

where n_1, n_2 are calculated by (25) and (26), respectively.

The slope of right wedge wave is $s_{\text{slope-right}} = (\text{floor}(4 \times M_{\text{horiz}}) + 1 - W_{\text{mid}}(1))/\text{floor}(4 \times M_{\text{vert}})$.

The middle line matrix $M_{\text{mid-line-right}} = (a_{n_1,n_2})_{s_{\text{wedge}}\times W_{\text{wedge}}}$, where $a_{n_1,n_2} = W_{\text{mid}}(1) + s_{\text{slope-right}} \times (Y_{\text{wrap}}(n_1, n_2) - 1)$.

The right coordinate matrix is $C_{\text{cord-right}} = (a_{n_1,n_2})_{s_{\text{wedge}}\times W_{\text{wedge}}}$, where

$$a_{n_1,n_2} = 0.5 + \frac{\text{floor}\left(4 \times M_{\text{vert}}\right)}{W_{\text{end}}(2) - W_{\text{end}}(1)} \times \frac{X_{\text{wrap}}\left(n_1, n_2\right) - M_{\text{mid-line-right}}\left(n_1, n_2\right)}{\text{floor}\left(4 \times M_{\text{vert}}\right) + 1 - Y_{\text{wrap}}\left(n_1, n_2\right)},$$

$$c2 = 1 \times \left(\frac{1}{2 \times \left(\text{floor}\left(4 \times M_{\text{horiz}}\right)/\left(W_{\text{end}}(1) - 1\right)\right) - 1} - 1 + 1 \times \left(2 \times \left(\frac{\text{floor}\left(4 \times M_{\text{vert}}\right)}{r_{\text{first-wedge-endpont}} - 1}\right) - 1\right)^{-1}\right)^{-1},$$

$$c1 = \frac{c2}{2 \times \left(\left(\text{floor}\left(4 \times M_{\text{horiz}}\right)\right)/\left(r_{\text{first-wedge-endpont}} - 1\right)\right) - 1}. \tag{31}$$

The corner coordinate matrix $C_{\text{cord-corner}} = (a_{n_1,n_2})_{s_{\text{wedge}}\times W_{\text{wedge}}}$, where

$$a_{n_1,n_2} = c1 + c2 \times \left[\frac{X_{\text{wrap}}\left(n_1, n_2\right) - 1}{\text{floor}\left(4 \times M_{\text{horiz}}\right)} - \frac{Y_{\text{wrap}}\left(n_1, n_2\right) - 1}{\text{floor}\left(4 \times M_{\text{vert}}\right)}\right] \times \left(2 - \left[\frac{X_{\text{wrap}}\left(n_1, n_2\right) - 1}{\text{floor}\left(4 \times M_{\text{horiz}}\right)} - \frac{Y_{\text{wrap}}\left(n_1, n_2\right) - 1}{\text{floor}\left(4 \times M_{\text{vert}}\right)}\right]\right)^{-1}. \tag{32}$$

The $C_{\text{cord-corner}}$ is wrapped, yielding the matrix $W_{l\text{-left}}$ and $W_{r\text{-right}}$:

$$W_{l\text{-left}}\left(n_1, n_2\right) = \begin{cases} 1, & C_{\text{cord-coner}}\left(n_1, n_2\right) = 0, \\ e^{(1-1/(1-e^{(1-1/C_{\text{cord-corner}}(n_1,n_2))}))}, & 0 < C_{\text{cord-corner}}\left(n_1, n_2\right) < 1, \\ 0, & C_{\text{cord-}l\text{corner}}\left(n_1, n_2\right) = 1, \end{cases}$$

$$W_{l\text{-right}}\left(n_1, n_2\right) = \begin{cases} 1, & C_{\text{cord-corner}}\left(n_1, n_2\right) = 0, \\ e^{(1-1/(1-e^{(1-1/(1-C_{\text{cord-corner}}(n_1,n_2)))}))}, & 0 < C_{\text{cord-corner}}\left(n_1, n_2\right) < 1, \\ 0, & C_{\text{cord-corner}}\left(n_1, n_2\right) = 1. \end{cases} \tag{33}$$

In the same way, the $C_{\text{cord-right}}$ is wrapped, yielding the matrix $W_{wl\text{-left}}$ and $W_{wr\text{-right}}$.

The discrete locating window of wedge wave W_{data} is filtered, yielding the matrix $W_{\text{data-tran}} = (a_{n_1,n_2})_{s_{\text{wedge}}\times W_{\text{wedge}}}$, where $a_{n_1,n_2} = W_{\text{data}}(n_1, n_2) \times [W_{l\text{-left}}(n_1, n_2) \times W_{wr\text{-right}}(n_1, n_2)]$.

The matrix $W_{\text{data-tran}}$ is rotated, yielding the matrix $W_{\text{data_2}}$

$$W_{\text{data_2}} = \text{rot}\,90\left(W_{\text{data-tran}}, -\left(\text{quad} - 1\right)\right), \tag{34}$$

where $\text{rot}\,90(A, k)$ rotates matrix A counterclockwise by $k \times 90^\circ$ degrees.

(7) Inverse 2DFFT is applied to $W_{\text{data_2}}$, hence generating the Curvelet coefficient at scale 4 and angle 1, $C_{4,1}$.

(8) Repeat (5), (6), and (7) in Step 3, in the same way of acquiring $C_{4,1}$; Curvelet coefficients at scale 4 and angle from 2 to 8 are generated.

Noted that at angle from 2 to 32, the left line vector is denoted by $s_{\text{left_line}} = (a_{1,k})_{1\times(s_{\text{wedge}})}$, where $a_{1,k} = \text{round}(W_{\text{end}}(l_{\text{sub}} - 1) + s_{\text{slope}} \times (k - 1)) \cdots k = 1, 2, \ldots, s_{\text{wedge}}$. Consider

$$l_{\text{sub}} = 2, 3, 4, \ldots, 8. \tag{35}$$

Table 1: Structure of Curvelet transform coefficients.

Levels	Scales (j)	Orientation (l)	Matrix form $C_{j,i}$
Coarse	1	1	21×21
Details	2	16	18×22 16×22 16×22 18×22 22×18 22×16 22×16 22×18 18×22 16×22 16×22 18×22 22×18 22×16 22×16 22×18
	3	32	34×22 32×22 32×22 32×22 32×22 32×22 32×22 34×22 22×34 22×32 22×32 22×32 22×32 22×32 22×32 22×34 34×22 32×22 32×22 32×22 32×22 32×22 32×22 34×22 22×34 22×32 22×32 22×32 22×32 22×32 22×32 22×34
	4	32	67×44 64×43 64×43 64×44 64×44 64×43 64×43 67×44 44×67 43×64 43×64 44×64 44×64 43×64 43×64 44×67 67×44 64×43 64×43 64×44 64×44 64×43 64×43 67×44 44×67 43×64 43×64 44×64 44×64 43×64 43×64 44×67
Fine	5	1	256×256

(9) The Curvelet coefficients at scale 4 and other three quadrants, $C_{4,l}$, $l = 9, 10, \ldots, 32$ are acquired in the same way as that in the first quadrants.

Finally, the Curvelet coefficients at scale 4, $C_{4,l}$, $l = 1, 2, \ldots, 32$, are generated after Step 3.

Step 4. Repeat Step 3, hence generating the Curvelet coefficients at scale 3, $C_{3,l}$, $l = 1, 2, \ldots, 32$.

Note that the discrete locating window of wedge wave at scale 3 can be calculated by

$$W_{\text{data_3}} = \left(a_{n_1,n_2}\right)_{s_{\text{wedge-3}} \times W_{\text{wedge-3}}}, \tag{36}$$

where

$$a_{n_1,n_2} = \begin{cases} f'_{\text{hipass}_3}(n_1, n_2), & V_{\text{col}_3}(n_2) > 0, \\ 0, & V_{\text{col_3}}(n_2) \le 0, \end{cases} \tag{37}$$

$s_{\text{wedge-3}}$, $W_{\text{wedge-3}}$ is the length and width of the discrete locating window of wedge wave at scale 3, and $V_{\text{col_3}}$ is condition vector at scale 3.

Step 5. Repeat Step 3, hence generating the Curvelet coefficients at scale 2, $C_{2,l}$, $l = 1, 2, \ldots, 16$.

Note that the discrete locating window of wedge wave at scale 2 can be calculated by

$$W_{\text{data_2}} = \left(a_{n_1,n_2}\right)_{s_{\text{wedge-2}} \times W_{\text{wedge-2}}}, \tag{38}$$

where

$$a_{n_1,n_2} = \begin{cases} f'_{\text{hipass}_2}(n_1, n_2), & V_{\text{col}_2}(n_2) > 0, \\ 0, & V_{\text{col_2}}(n_2) \le 0, \end{cases} \tag{39}$$

and $V_{\text{col_2}}$ is condition vector at scale 2.

Step 6. Inverse 2DFFT is applied to the low-pass signal at scale 2, $f'_{\text{lowpass}_2}(n_1, n_2)$, generating the Curvelet coefficient at scale 1 $C_{1,1}$.

The detailed structure of the Curvelet coefficients obtained by FDCT_WARPING is shown in Table 1.

2.3.2. Fingerprint Image Noise Filtration Technique. Noise always arises from the acquiring fingerprint images. The noise may result in the vagueness and many discontinuities of ridges (or valleys) in the image and thus affects accurate feature extraction and recognition. So, it is necessary and important to denoise in fingerprint images.

The relationship of the Curvelet coefficients between the different scales is similar to the relationship of the wavelet coefficients; that is, there exists strong correlation between them.

From Table 1, there are 16 and 32 orientations at scale 2 and scale 3, respectively. Each Curvelet coefficient matrix is at scale 2 and each orientation corresponds to two adjacent matrices generated at scale 3. The ridges in a fingerprint image correspond to the Curvelet coefficients with large magnitude at scale 2. Each Curvelet coefficient matrix at scale 2 is decomposed into two Curvelet coefficient matrices at scale 3 and at two adjacent orientations. The corresponding two Curvelet coefficient matrices at scale 3 also have large magnitude, while the magnitude of the Curvelet coefficients corresponding to the noise dies out swiftly from scale 2 to scale 3. So, we use the direct spatial correlation of Curvelet coefficients at scale 2 and scale 3 to accurately distinguish ridges from noise. For scales 4 and 5, we adopt hard thresh method to filter the noise. Finally, we reconstruct all the Curvelet coefficients by the technique [17] and accomplish fingerprint image filtration.

The proposed noise filtration technique has the following steps.

Step 1. Noise filtration of Curvelet coefficient matrices at scale 2 and scale 3, respectively.

This section details the major steps of the proposed noise filtration algorithm of Curvelet coefficient matrices generated at scales 2 and 3.

Assume each matrix to be at scale 2 and orientation l, $C_{2,l}$, corresponds to two adjacent matrices generated at scale 3, $C_{3,2\times(l-1)+1}$ and $C_{3,2\times(l-1)+2}$, (l ranges from 1 to $16\times2^{\mathrm{ceil}((5-3)/2)}$). Let (r_0, c_0) be the size of $C_{2,l}$. The sizes of $C_{3,2\times(l-1)+1}$ and $C_{3,2\times(l-1)+2}$ are $(r_1 \times c_1)$ and $(r_2 \times c_2)$, respectively. The matrices at scales 2 and 3 are filtered as follows.

(1) Decompose the Curvelet coefficients matrix $C_{3,2\times(l-1)+1}$ into two matrices A and B with the same size as $C_{2,l}$, $r_0 \times c_0$. The elements of A are extracted from row 1 to r_0 and column 1 to c_0 of $C_{3,2\times(l-1)+1}$, and the elements of B are extracted from row $r_1 - r_0 + 1$ to r_1 and column $c_1 - c_0 + 1$ to c_1 of $C_{3,2\times(l-1)+1}$. Consider

$$A = \left(a_{n_1,n_2}\right)_{r_0\times c_0}, \tag{40}$$

where $a_{n_1,n_2} = C_{3,2\times(l-1)+1}(n_1, n_2)$, $n_1 = 1, 2, 3, \ldots, r_0$, $n_2 = 1, 2, 3, \ldots, c_0$, and

$$B = \left(a_{n_1,n_2}\right)_{r_0\times c_0}, \tag{41}$$

where $a_{n_1,n_2} = C_{3,2\times(l-1)+1}(n_1, n_2)$, $n_1 = r_1 - r_0 + 1$, $r_1 - r_0 + 2, \ldots, r_1$, $n_2 = c_1 - c_0 + 1, c_1 - c_0 + 2, \ldots, c_1$.

(2) Decompose the Curvelet coefficients matrix $C_{3,2\times(l-1)+2}$ into two matrices D and E with the size of $r_0 \times c_0$:

$$D = \left(a_{n_1,n_2}\right)_{r_0\times c_0}, \tag{42}$$

where $a_{n_1,n_2} = C_{3,2\times(l-1)+2}(n_1, n_2)$, $n_1 = 1, 2, 3, \ldots, r_0$, $n_2 = 1, 2, 3, \ldots, c_0$, and

$$E = \left(a_{n_1,n_2}\right)_{r_0\times c_0}, \tag{43}$$

where $a_{n_1,n_2} = C_{3,2\times(l-1)+2}(n_1, n_2)$, $n_1 = r_1 - r_0 + 1$, $r_1 - r_0 + 2, \ldots, r_1$, $n_2 = c_1 - c_0 + 1, c_1 - c_0 + 2, \ldots, c_1$.

(3) Calculate the four multiplication coefficient matrices (m_k, $k = 1, 2, 34$) between $C_{2,l}$ and the four matrices A, B, D, and E, respectively. Consider

$$\begin{aligned}
&m_1(n_1, n_2) = C_{2,l}(n_1, n_2) \times A(n_1, n_2),\\
&\quad n_1 = 1, 2, \ldots, r_0, \quad n_2 = 1, 2, \ldots, c_0,\\
&m_2(n_1, n_2) = C_{2,l}(n_1, n_2) \times B(n_1, n_2),\\
&\quad n_1 = 1, 2, \ldots, r_0, \quad n_2 = 1, 2, \ldots, c_0,\\
&m_3(n_1, n_2) = C_{2,l}(n_1, n_2) \times D(n_1, n_2),\\
&\quad n_1 = 1, 2, \ldots, r_0, \quad n_2 = 1, 2, \ldots, c_0,\\
&m_4(n_1, n_2) = C_{2,l}(n_1, n_2) \times E(n_1, n_2),\\
&\quad n_1 = 1, 2, \ldots, r_0, \quad n_2 = 1, 2, \ldots, c_0.
\end{aligned} \tag{44}$$

(4) Filter the Curvelet coefficient matrices $C_{3,2\times(l-1)+1}$ and $C_{3,2\times(l-1)+2}$ by

$$\begin{aligned}
&A'(n_1, n_2)\\
&= \begin{cases} A(n_1, n_2), & \left||m_1(n_1, n_2)| - |m_3(n_1, n_2)|\right| > T_s,\\ 0, & \text{otherwise,}\end{cases}\\
&D'(n_1, n_2)\\
&= \begin{cases} D(n_1, n_2), & \left||m_1(n_1, n_2)| - |m_3(n_1, n_2)|\right| > T_s,\\ 0, & \text{otherwise,}\end{cases}\\
&B'(n_1, n_2)\\
&= \begin{cases} B(n_1, n_2), & \left||m_2(n_1, n_2)| - |m_4(n_1, n_2)|\right| > T_s,\\ 0, & \text{otherwise,}\end{cases}\\
&E'(n_1, n_2)\\
&= \begin{cases} E(n_1, n_2), & \left||m_2(n_1, n_2)| - |m_4(n_1, n_2)|\right| > T_s,\\ 0, & \text{otherwise,}\end{cases}
\end{aligned} \tag{45}$$

where A', D', B', and E' are the Curvelet coefficient matrices at scale 3 after filtration, the operation $|\cdot|$ returns the complex modulus (magnitude) of the input, and T_s is the threshold. For example, if $||m_1(n_1, n_2)| - |m_3(n_1, n_2)|| > T_s$, we assume that the Curvelet coefficients correspond to the ridges of the image. Otherwise, the Curvelet coefficients correspond to the noise of the image, which are assigned to 0.

(5) When filtering the Curvelet coefficient matrix $C_{2,l}$, if any of the filtered Curvelet coefficient matrices (A', D', B',and E') equal to Z, $C_{2,l}$ is considered as noise and assigned to Z. Where Z is a matrix with all elements are zero.

(6) repeat (1) to (5) with 16 times (the number of orientation of Curvelet coefficients at scale 2).

After (1)–(5), all the Curvelet coefficients at scale 2 and scale 3 are filtered.

Step 2. Noise Filtration of Curvelet coefficient matrices at scale 4 and scale 5, respectively.

The Curvelet coefficient matrices generated at scale 4 are filtered by

$$C'_{j,l}(i, j) = \begin{cases} C_{j,l}(n_1, n_2), & \left|C_{j,l}(n_1, n_2)\right| > \mathrm{thresh} \times E_{j,l},\\ 0, & \text{otherwise,}\end{cases} \tag{46}$$

where $C'_{j,l}$ are the filtered Curvelet coefficient matrices, $|\cdot|$ is the complex modulus operation, and thresh is the threshold; in this paper, thresh = 1.5, and

$$E_{j,l} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sqrt{\left|C_{j,l}(i,j)\right|^2}}{n\times m}, \tag{47}$$

where (n, m) is the size of matrix $C_{j,l}$.

The Curvelet coefficient matrices generated at scale 5 are filtered in the same way as the Curvelet coefficient matrices generated at scale 4. Note that at scale 5 thresh = 2.

Step 3. After the coefficients at scale 2, 3, 4, and 5 are filtered, we reconstruct the coefficients using the technique [16] and accomplish image noise filtration.

T_s in (45) can be acquired by the statistics of difference of correlation coefficients in adjacent directions at the same scale. Finally, $T_s = 100$ is selected by many experiments. Figure 1 shows the noise filtration results of five types of fingerprint image by proposed noise filtration algorithm.

From Figure 1 (x):(B) (x = a, b, c, d, e), we can see many discontinuities of ridges in the original image can be smoothed after filtering and the direction of ridge is well followed, which founds the good basis for accurate feature extraction and recognition. We demonstrate Curvelet coefficients at different scales of five types of filtered images in Figure 2.

As Figure 2(a) to Figure 2(e) show, there are strong orientations in the Curvelet coefficients images. The white parts in the images represent partial edges of the ridge of fingerprint image in different orientations. Meanwhile, it means the significant Curvelet coefficients of images. The low-frequency (coarse scale) coefficients are stored at the center of the display. The Cartesian concentric coronae show the coefficients at different scales; the outer coronae correspond to higher frequencies. There are four strips associated with each corona, corresponding to the four cardinal points; these are further subdivided in angular panels. Each panel represents coefficients at a specified scale and along the orientation suggested by the position of the panel.

2.4. Fingerprint Feature Extraction. Haralick et al. [18] first proposed gray-level cooccurrence matrix (GLCM) for texture descriptions in the 1970s. It is still popular until today and widely used in various texture classifications [19–23], because of its good statistic performance. The GLCM is a second order statistics method which describes the spatial interrelationships of the grey tones in an image.

GLCM contains elements that are counts of the number of pixel pairs, which are separated by certain distance and at some angular direction. Typically, the GLCM is calculated in a small window, which scans the whole image. The texture feature will be associated with each pixel.

In our studies, GLCM is computed based on two parameters, which are the distance between the pixel pair d and their angular relationship θ. $d = 1$ and θ are quantized in four directions (0°, 45°, 90°, and 135°). For image I, defined square window $N \times N$, brightness levels i and j, the nonnormalized GLCM p_{ij} are defined by

Table 2: Direction values.

θ	0°	45°	90°	135°
θ_0	0	1	1	1
θ_1	1	1	0	−1

$$p_{i,j,\theta} = \sum_{x=1}^{N}\sum_{y=1}^{N} C\left\{\left(I(x,y)=i\right)^{(I(x\pm d\theta_0, y\mp d\theta_1)=j)}\right\}, \tag{48}$$

where $C\{\cdot\} = 1$ if the argument is true and $C\{\cdot\} = 0$, otherwise. The signs $\pm$ and $\mp$ in (10) mean that each pixel pair is counted twice: once forward and once backward in order to make the GLCM diagonally symmetric. For each direction, θ_0 and θ_1 are shown in Table 2.

The procedures of feature extraction are as follows.

Step 1. Scale the grayscale values in Curvelet transform coefficients into 8 levers and compute the GLCMs of Curvelet coefficients at scale 1 $C_{1,1}$ and calculate 16 texture features based on 4 GLCMs:

(1) Angular second moment (ASM)

$$f_1 = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p(i,j)^2. \tag{49}$$

(2) Contrast (CON)

$$f_2 = \sum_{n=0}^{G-1} n^2 \left\{\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p(i,j)^2\right\}, \tag{50}$$

where $|i - j| = n$.

(3) Correlation (COR)

$$f_3 = \frac{\sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(ij)\,p(i,j) - \mu_1\mu_2}{\sigma_1^2\sigma_2^2}, \tag{51}$$

where

$$\mu_1 = \sum_{i=0}^{G-1} i\sum_{j=0}^{G-1} p(i,j), \qquad \mu_2 = \sum_{i=0}^{G-1} j\sum_{j=0}^{G-1} p(i,j),$$

$$\sigma_1^2 = \sum_{i=0}^{G-1}(i-\mu_1)^2\sum_{j=0}^{G-1} p(i,j), \tag{52}$$

$$\sigma_2^2 = \sum_{i=0}^{G-1}(j-\mu_2)^2\sum_{i=0}^{G-1} p(i,j).$$

(4) Entropy (ENT)

$$f_4 = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p(i,j)\log p(i,j). \tag{53}$$

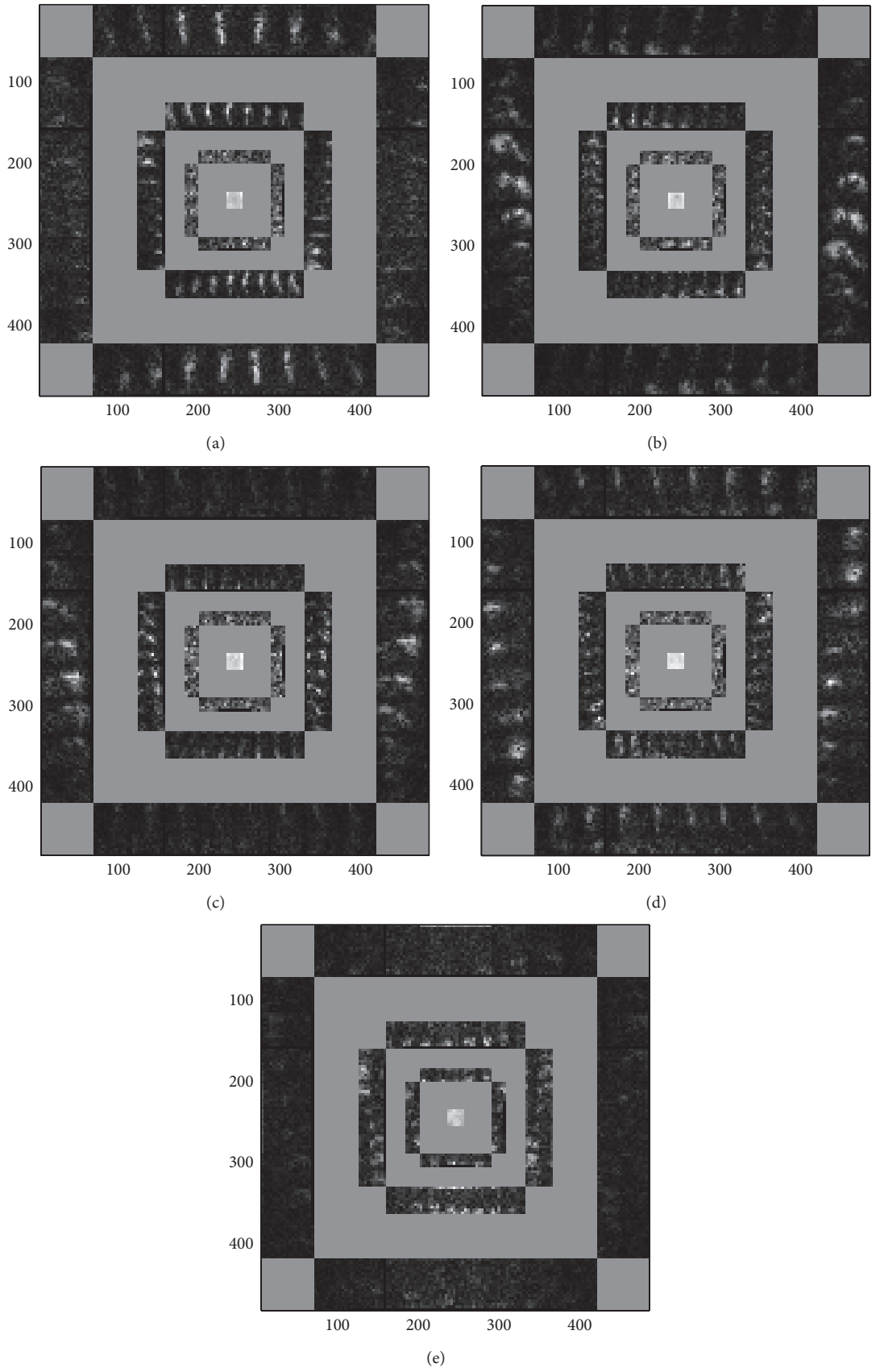


Figure 2: Curvelet coefficients of five filtered fingerprint classes. (a) Right loop (R), (b) left loop (L), (c) whorl (W), (d) arch (A), and (e) tented arch (T).

Step 2. Calculate averaged l_1-norm of Curvelet coefficients in 8 directions at the second scale $C_{2,l}$ and acquire 8 texture features according to (51). Consider

$$E_l = \frac{1}{m_1 \times m_2} \sum_{i}^{m_1} \sum_{j}^{m_2} \left| C_{2,l}(i,j) \right|, \tag{54}$$

where, (m_1, m_2) is the size of the matrix $C_{2,l}$, and $l = 1, 2, 3, 4, 5, \ldots, 16$.

Step 3. Calculate averaged l_1-norm of Curvelet coefficients in 16 directions at the third scale $C_{3,l}$ and acquire 16 texture features according to (54).

Step 4. Calculate averaged l_1-norm of Curvelet coefficients in 16 directions at the fourth scale $C_{4,l}$ and acquire 16 texture features according to (54).

Step 5. Calculate averaged l_1-norm of Curvelet coefficients at the fifth scale $C_{5,1}$ and acquire 1 texture feature.

Note that in Step 2, Step 3, and Step 4, we only calculate the averaged l_1-norm of Curvelet coefficients in even directions to assure the classification accuracy and reduce the recognition time. So, a feature vector containing 57 components for each image can be extracted.

3. Experiment Results

3.1. Datasets. NIST special fingerprint database 4(NIST-4) is one of the most important benchmarks for fingerprint classification. Most published results on fingerprint classification are based on this database. For comparison with other approaches, we also perform our fingerprint classification algorithm on this database for the five-class fingerprint classification problem. Since fingerprint classes A (arch) and T (tented arch) have a substantial overlap, it is very difficult to separate these two classes. Therefore, we also report our results for the four-class classification problem, where classes A and T have been merged into one class. NIST-4 contains 4000 fingerprints of size 480 × 512 pixels, taken from 2000 fingers. Each finger has two impressions (f and s). The first fingerprint instances are numbered from f0001 to f2000 and the second fingerprint instances are numbered from s0001 to s2000. All fingerprints in this database are used in our experiment. We form our training set with the first 2,000 fingerprints from 1,000 fingers (f0001 to f1000 and s0001 to s1000) and the test set contains the remaining 2,000 fingerprints (f1001 to f2000 and s1001 to s2000).

To eliminate the large difference between the feature vectors, each feature vector v_k is normalized according to

$$v_k(i) = \frac{v_k(i) - v_{\min}(i)}{v_{\max}(i) - v_{\min}(i)}, \tag{55}$$

where $v_k(i)$ represents the ith element of vector v_k, $v_{\min}(i)$ denotes the minimum of the ith element of all the row vectors, and $v_{\max}(i)$ denotes the maximum of the ith element of all the row vectors.

Table 3: Fingerprint classification results on NIST-4.

True class	Hypothesized class				
	A	T	L	R	W
A	432	12	3	1	2
T	28	296	8	4	0
L	4	5	370	0	4
R	3	8	1	386	3
W	3	1	6	5	380

3.2. Experiment Results and Analysis. The performance of a fingerprint classification algorithm is often measured in terms of accuracy. The accuracy is computed as the ratio between the number of correctly classified fingerprint and the total number of fingerprints in the test set. Each image is labeled with one or more of the five classes (W, R, L, A, and T). To simplify the training procedure, we make use of only the first label of a fingerprint to train our system. For testing, however, we make use of all the labels for a fingerprint and consider the output of our classifier to be correct if the output matches any one of the labels. This is in line with the common practice used by other researchers in comparing the classification results on the NIST-4 database.

Classification accuracy does not always increase with increasing K of the K-nearest neighbor classifier; there exists an optimal value of K for finite training sample size classification problems. According to the method in [24], in our experiments, 10 nearest neighbors ($K = 10$) are considered. The classification results of our proposed approach are shown in Table 3. The diagonal entries in Table 3 show the number of test patterns from different classes which are correctly classified.

From Table 3, we can conclude that the proposed algorithm achieves an accuracy of 94.6 percent for the five-class classification task. For the four-class classification task (where classes A and T were collapsed into one class), an accuracy of 96.8 percent is achieved.

Experiment. To evaluate the performance of the proposed algorithm, we have compared the proposed approach to wavelet-based, GLCM-based, and Curvelet-based, respectively. We use wavelet transform to decompose gray images into five scales wavelet coefficients using wavelet bases "Symmlets 4, 5, 6, 8, and 9" and calculate averaged l_1-norm of wavelet coefficients at each scale. Finally, WT feature vector with dimension of 16 are acquired. The reason using "Symmlets 4, 5, 6, 8, and 9" is that in the work of Tico et al. [10] best results were obtained with these five wavelet bases. In Table 4, we show the comparison results for the five-class classification.

From Table 4, we can conclude that our algorithm achieves higher accuracy of classes W by reducing the misclassification of W as L or R. Our algorithm also achieves higher accuracy of classes R by reducing the misclassification of R as A. Finally, our algorithm achieves higher accuracy of classes A by reducing the misclassification of A as T.

Table 4: Comparison of accuracy on NIST-4 for the five-class classification.

Features	Accuracy A	Accuracy T	Accuracy L	Accuracy R	Accuracy W
Wavelet (Symmlet 4)	92%	79.7%	94.2%	93.3%	93.7%
Wavelet (Symmlet 5)	92%	82.7%	94.5%	93.5%	93.1%
Wavelet (Symmlet 6)	92%	83.7%	94.2%	93.0%	93.7%
Wavelet (Symmlet 8)	92%	82.7%	94.2%	93.0%	93.7%
Wavelet (Symmlet 9)	92%	81.2%	94.2%	93.0%	93.7%
GLCM	78.4%	77.7%	79.4%	72.8%	82.7%
Curvelet	93.6%	84.7%	95.0%	94.0%	94.5%
Proposed algorithm	96.0%	88.1%	96.6%	96.3%	96.2%

Also, Curvelet-based is better than wavelet-based and GLCM-based. The reason is CT can better capture the direction of fingerprint ridge than WT and GLCM. Furthermore, the proposed algorithm can provide much more information on the ridge direction by combining the good statistic performance of GLCM and well capturing the direction of CT.

Most of misclassifications in the proposed approach are caused by heavy noise in the poor quality fingerprints, where it is very difficult to correctly extract Curvelet coefficients.

4. Conclusion

In this paper, we present an efficient fingerprint classification algorithm that uses CT and GLCM to model the feature set of fingerprint. There are two main contributions in this paper. Firstly, we construct Curvelet filter that can smooth the discontinuities of ridges and remove the noise in the original image. As a result, the direction of ridge is well followed. Secondly, in combination with the effectiveness of CT and GLCM, we propose to construct a 53-dimensional feature vector as classifier input that can represent curves singularities and the statistics in fingerprint image with compact feature. We have tested our algorithm on the NIST-4 database and a very good performance has been achieved (94.6 percent for the five-class classification problem and 96.8 percent for the four-class classification problem with 1.8 percent rejection). These good performances of the proposed algorithm could be ascribed to the high information contents of Curvelet features and to the combination of GLCM and CT.

Our system takes about 1.47 seconds on a AMD E-350 PC to classify one fingerprint, which needs to be improved. Since image decomposition (filtering) steps account for 82 percent of the total compute time, special purpose hardware for Curvelet transform can significantly decrease the overall time for classification.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful for the anonymous reviewers who made constructive comments. This work is supported by the National Natural Science Foundation of China (no. 61203302 and no. 51107088), the Tianjin Research Program of Application Foundation, and Advanced Technology (14JCYBJC18900).

References

[1] J.-M. Guo, Y.-F. Liu, J-Y. Chan, and J.-D. Lee, "Fingerprint classification based on decision tree from singular points and orientation field," *Expert Systems with Applications*, vol. 41, no. 2, pp. 752–764, 2014.

[2] D. Peralta, I. Triguero, R. Sanchez-Reillo, F. Herrera, and J. M. Benitez, "Fast fingerprint identification for large databases," *Pattern Recognition*, vol. 47, no. 2, pp. 588–602, 2014.

[3] M. Liu, "Fingerprint classification based on Adaboost learning from singularity features," *Pattern Recognition*, vol. 43, no. 3, pp. 1062–1070, 2010.

[4] H. Ching-Tang and H. Chia-Shing, "Fingerprint identification based on MOPSO in SVM," *Applied Mechanics and Materials*, vol. 479-480, pp. 810–817, 2014.

[5] K. Cao, L.-J. Pang, J.-M. Liang, and J. Tian, "Fingerprint classification by a hierarchical classifier," *Pattern Recognition*, vol. 46, no. 12, pp. 3186–3197, 2013.

[6] M. Liu and P. Yap, "Invariant representation of orientation fields for fingerprint indexing," *Pattern Recognition*, vol. 45, no. 7, pp. 2532–2542, 2012.

[7] F. Liu, Q.-J. Zhao, and D. Zhang, "A novel hierarchical fingerprint matching approach," *Pattern Recognition*, vol. 44, no. 8, pp. 1604–1613, 2011.

[8] M. U. Munir, M. Y. Javed, and S. A. Khan, "A hierarchical k-means clustering based fingerprint quality classification," *Neurocomputing*, vol. 85, no. 15, pp. 62–67, 2012.

[9] A. K. Jain, S. Prabhakar, and L. Hong, "A Multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.

[10] M. Tico, P. Kuosmanen, and J. Saarinen, "Wavelet domain features for fingerprint recognition," *Electronics Letters*, vol. 37, no. 1, pp. 21–22, 2001.

[11] C.-B. Tao and G.-D. Liu, "Fingerprint image enhancement algorithm based on FDCT," *Advanced Materials Research*, vol. 255–260, pp. 2047–2051, 2011.

[12] M.-H. Liu, "Fingerprint classification based on Adaboost learning from singularity features," *Pattern Recognition*, vol. 43, no. 3, pp. 1062–1070, 2010.

[13] T. Mandal and Q. M. J. Wu, "A small scale fingerprint matching scheme using digital curvelet transform," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 1534–1538, October 2008.

[14] J. Liu and W. L. Wang, "A fingerprint matching algorithm based on reference point of central area," *Computer Applications and Software*, vol. 28, no. 2, pp. 30–33, 2011.

[15] E. J. Candese and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Communications on Pure and Applied Mathematics*, vol. 57, no. 2, pp. 219–266, 2004.

[16] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Journal of Multiscale Modeling and Simulation*, vol. 5, no. 3, pp. 861–899, 2006.

[17] http://www.curvelet.org/download/download.html.

[18] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[19] S. Hu, C. Xu, W.-Q. Guan, Y. Tang, and L. Yan, "Texture feature extraction based on wavelet transform and gray-level co-occurrence matrices applied to osteosarcoma diagnosis," *Bio-Medical Materials and Engineering*, vol. 24, no. 1, pp. 129–143, 2014.

[20] V. S. Thakare and N. N. Patil, "Classification of texture using gray level co-occurrence matrix and self-organizing map," in *Proceedings of the International Conference on Electronic Systems, Signal Processing, and Computing Technologies*, pp. 350–355, 2014.

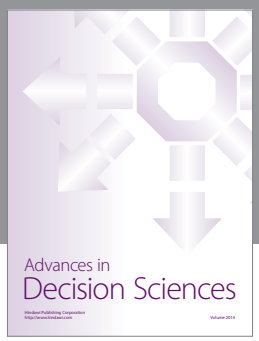
[21] A. Suresh and K. L. Shunmuganathan, "Feature fusion technique for colour texture classification system based on gray level co-occurrence matrix," *Journal of Computer Science*, vol. 8, no. 12, pp. 2106–2111, 2012.

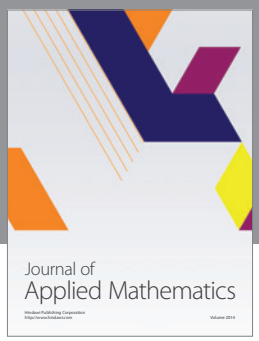
[22] R. A. Pramunendar, C. Supriyanto, D. H. Novianto, I. N. Yuwono, G. F. Shidik, and P. N. Andono, "A classification method of coconut wood quality based on Gray Level Co-occurrence matrices," in *Proceedings of the IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (ROBIONETICS '13)*, pp. 254–257, Yogyakarta, Indonesia, November 2013.

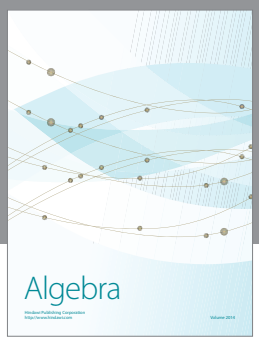
[23] M.-D. Bi, Z.-G. Sun, and Y.-S. Li, "Textural fabric defect detection using adaptive quantized Gray-level Co-occurrence Matrix and Support Vector description data," *Information Technology Journal*, vol. 11, no. 6, pp. 673–685, 2012.

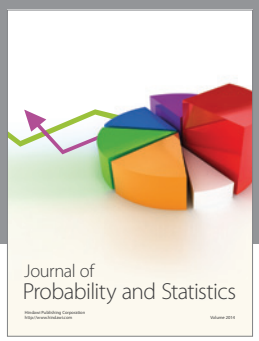
[24] L. Wang and M. Dai, "Application of a new type of singular points in fingerprint classification," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1640–1650, 2007.



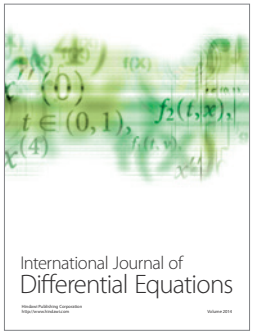


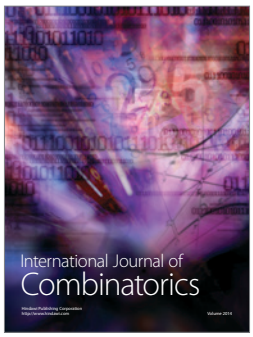


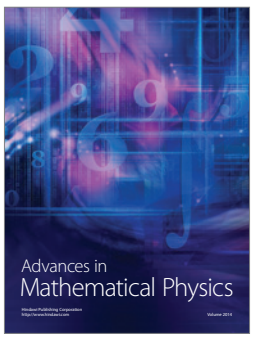


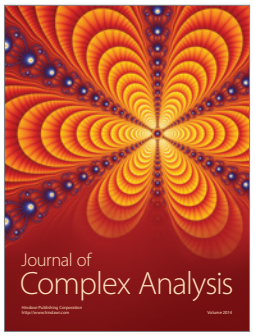




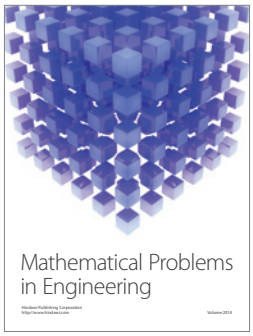


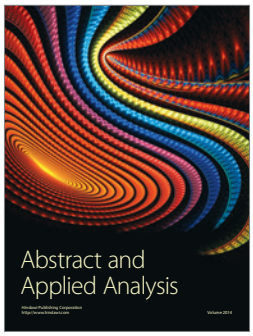


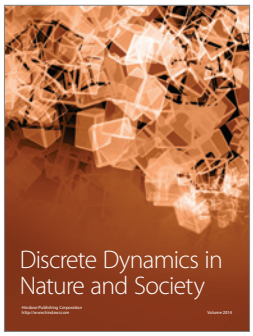


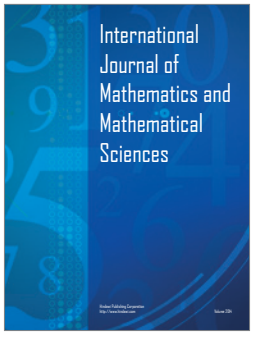






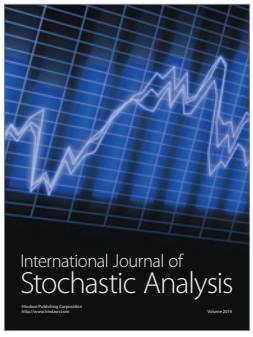














Hindawi
Submit your manuscripts at
http://www.hindawi.com