Efficient rotation- and scale-invariant texture analysis

Kam-Keung Fung
Kin-Man Lam
The Hong Kong Polytechnic University
Department of Electronic and Information Engineering
Centre for Signal Processing
Hung Hom, Kowloon
Hong Kong
E-mail: enkmlam@polyu.edu.hk

Abstract. *Texture analysis plays an important role in content-based image retrieval and other areas of image processing. It is often desirable for the texture classifier to be rotation and scale invariant. Furthermore, to enable real-time usage, it is desirable to perform the classification efficiently. Toward these goals, we propose several enhancements to the multiresolution Gabor analysis. The first is a new set of kernels called Slit, which can replace Gabor wavelets in applications where high computational speed is desired. Compared to Gabor, feature extraction using Slit requires only 11 to 17% of the numeric operations. The second is to make the features more rotation invariant. We propose a circular sum of the feature elements from the same scale of the feature vector. This has the effect of averaging the feature vector from all orientations. The third is a slide-matching scheme for the final stage of the classifier, which can be applied to different types of distance measures. Distances are calculated at slightly different scales, and the smallest value is used as the actual distance measures. Experimental results using different image databases and distance measures show distinct improvements over existing schemes. © 2010 SPIE and IS&T.* [DOI: 10.1117/1.3495999]

1 Introduction

All real-world objects have textures. When these objects are captured in 2-D digital images, their associated textural information provides important clues for the successful indexing, classification, and retrieval of these images. Image management based on texture analysis has already been adopted in some applications.[1] To make texture analysis useful for a wider range of practical applications, however, we need to further improve the performance in several areas. The first such area is the invariance to the rotation and scaling of images being classified or analyzed. A classifier that only performs well under no relative rotation and scaling of images will not perform well when real-world images are used. We propose enhancements to the general structure of the Gabor feature based on the circular sum of feature elements, and the use of slide matching across augmented scales for the calculation of distance measures.

The second area we want to contribute to is the speed of execution. Currently, most proposed feature extraction processes and classifiers perform well enough for certain prac-

tical applications. However, most, if not all, of them require significant computational resources. We have formulated a new set of kernels called Slit, which can replace the Gabor functions in the multiresolution analysis of texture, and an associated fast algorithm for the extraction of features. This can result in substantial reduction in computational requirements. Experimental results show a comparable classification performance between the Gabor feature and the Slit feature.

2 Literature Review

All enhancements proposed in this work are based on the multiresolution Gabor analysis. Numerous enhancements have been proposed in the literature to improve its rotation and scale invariance, and its computational efficiencies. The following is a brief survey.

2.1 Enhancements for Rotation and Scale Invariance

Lahajnar and Kovacic[2] tackle the rotation invariance problem via a fairly complicated scheme. An additional set of Gaussian filters is employed. The Fourier transform is applied to extract the Gabor energies, and histogram matching is used as the distance measure. The reported classification performance, on a small set of images, is on par with those reported in Ref. 3. Arivazhagan, Ganesan, and Priyal[4] improve the rotation invariance of Gabor analysis by first searching for the orientation with the largest energy at each scale. They circularly shift the associated means and standard deviations of these dominant orientations to the leading position of their associated scales. In effect, the image rotation is compensated by using the highest energy position as an indication of the rotated angle. This scheme can achieve significant improvements over traditional Gabor analysis, but incurs additional computation overheads, as the sorting and shifting of feature elements are necessary at each scale.

Alternatively, Han and Ma[5] improve rotation and scale invariance by summing the convoluted results with the same scale or orientation, before the means and standard deviations are calculated. The orientation or scale information is essentially averaged out. There are moderate penalties on computational efficiency, as the $N \times N$ convolution results need to be summed along the scale or orientation axis. Recently, Xie et al.[6] proposed to use an adaptive circular orientation normalization (ACON) scheme to improve rotation invariance,

and an elastic cross-frequency searching (ECFS) mechanism for improving scale invariance. The additional computation required is significant, involving the sorting of Gabor elements along each scale, and the searching of the minimum distance between all scales during matching. The reported performance on the classification of scaled and rotated images, however, is among the best in the literature.

Relatively few publications directly tackle the problem of scale-invariant object matching or texture classification. Kyrki, Kämäräinen, and Kälviäinen[7] propose a feature space that is based on the outputs of a set of Gabor filters at individual image locations. This feature space contains Gabor elements in rows and columns, and can be used for rotation- and scale-invariant matching. The accompanied algorithm describes a rudimentary scheme for searching of the best match in different rotations and scales. Rotation matching is performed through a circular match of the columns, quite similar to the circular shift method described in Ref. 4. Scale matching, on the other hand, is done by shifting the rows one by one. The proposed framework is only for individual points in the image, so it could be useful for object matching, like face and hands. There are numerous follow-up publications that cite this feature space, and quite a few of them utilize this space for face recognition and object tracking, but not texture analysis.

2.2 Addressing Computation Complexity

Generating features through Gabor analysis is computationally intensive. Even with the use of fast Fourier transform (FFT), this process still consumes significant processing time. Next are some of the more recent activities addressing this issue.

2.2.1 Exploiting structures of Gabor filters

Ilonen, Kämäräinen, and Kälviäinen[8] acknowledge that the computational complexity of multiresolution Gabor analysis needs to be improved before this effective scheme can be widely adopted for practical use. A series of simplifications, including the downsampling of images before filtering, using effective filter envelopes, and reusing filtering results are suggested. A similar but simpler approach is described earlier by Chen and Chen.[9] A rough estimation show a 58% savings in computational requirement. Retrieval results using a very small set of texture images are promising.

An earlier attempt by Namuduri, Mehrotra, and Ranganathan[10] takes a slightly different route. Gabor filters are restricted to octave spacing, and the filter output is generated recursively. By reusing results during these steps, they claim that the number of operations is reduced. Unfortunately, this study does not provide numeric figures for these possible improvements.

2.2.2 Approximating Gabor filters

Choi et al.[11] formulate a simple kernel, called the simplified Gabor wavelet (SGW). This family of kernels is an approximated version of the Gabor wavelets, and rectangles of different sizes and magnitudes are added together to mimic the Gabor kernels. Retrieval performances for both types of kernel are shown to be similar. The advantage of SGW is that a fast algorithm can be used for feature extraction, which can result in up to 35% less computation. This fast algorithm is based on the use of integral images and rotated integral images.

2.3 Multiresolution Gabor Analysis

The proposed algorithms are based on the framework due to Manjunath and Ma.[12] A 2-D Gabor function $g(x,y)$ can be expressed as

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right]. \quad (1)$$

This is the product of a 2-D Gaussian envelop, with its size determined by σ_x and σ_y and a complex plane wave with a frequency W.

For multiresolution analysis, a set of Gabor functions or kernels $g_{m,n}(x, y)$ can be generated by proper dilation and rotation of the mother kernel $g(x, y)$, through the generating function $g_{m,n}(x, y) = a^{-2m}g(x', y')$, where $a > 1$, $x' = a^{-m}(x\cos\theta_n + y\sin\theta_n)$, $y' = a^{-m}(-x\sin\theta_n + y\cos\theta_n)$, $\theta_n = n\pi/K$, $m = 0, 1, \ldots, S - 1$, and $n = 0, 1, \ldots, K - 1$. This set of Gabor kernels is said to contain S scales and K orientations. The scaling factor a^{-2m} is used to keep the overall *strength* of the filter consistent when its scale changes. A set of Gabor functions can form a complete but nonorthogonal basis set.

Since the Gabor kernel set is not orthogonal, there exists an infinite number of ways to have it constructed. In this work, we employ a fixed scaling factor, where a is set to a fixed value, normally $\sqrt{2}$, to effect an octave coverage.[13] The spreads of the Gaussian σ_x and σ_y are set to be equal, or $\sigma_x = \sigma_y = \sigma = \kappa/W$, where $\kappa = \sqrt{2\ln 2}[(2^\phi + 1)/(2^\phi - 1)]$, and ϕ is the overall spacing in the number of octaves.

The image being analyzed, $i(x, y)$, is filtered by each of the kernels in the set using the convolution operation $W_{m,n}(x, y) = \sum_{x1}\sum_{y1} i(x_1, y_1)g_{m,n}(x - x_1, y - y_1)$. Means and standard deviations of the filtered results from the whole kernel set are grouped together to form the Gabor feature $\bar{f}$ of the image, where $\bar{f} = \{\mu_{0,0}, \sigma_{0,0}, \mu_{0,1}, \sigma_{0,1}, \ldots, \mu_{S-1,K-1}, \sigma_{S-1,k-1}\}$. Individual entries in the Gabor feature are called Gabor elements of the image.

3 Proposed Algorithms

There are three parts to the proposed algorithms. We first look at the Slit kernel.

3.1 Slit Kernels

We introduce a simple kernel called Slit that can replace the Gabor kernel in texture analysis. Extraction of the Slit feature can be very efficient by making use of the kernel's geometrical properties in the spatial domain. Experimental results indicate that the Slit feature is just as effective as the Gabor feature in discriminating against different texture images. Furthermore, enhancements made to the Gabor feature for improving rotation and scale invariance can also be applied to the Slit feature.

The Slit kernels are approximations of the Gabor kernels, with the overall shapes simplified to rectangles, and all levels reduced to two magnitudes. Much like the Gabor kernels, the Slit kernels can also be scaled and rotated. They can thus act
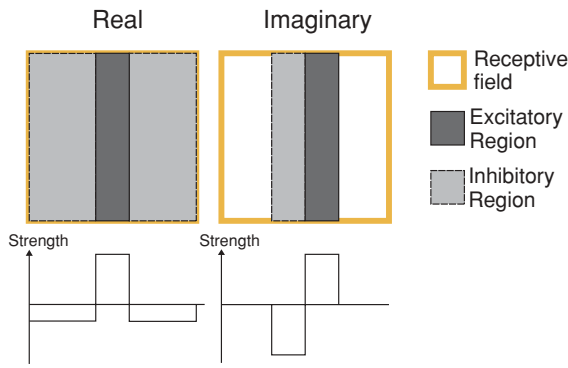
Real   Imaginary

Fig. 1 General structure of the Slit kernel.

as bandpass filters, with center frequencies similar to those for the Gabor kernels.

A complex set of Slit kernels is formulated, so the general framework of multiresolution Gabor analysis can be employed. This means two out-of-phase kernels are constructed simultaneously for each scale and orientation. The *real* part of these kernels, denoted Slit$_\text{real}$, is constructed as even symmetric, while the *imaginary* part, denoted Slit$_\text{imag}$, is made odd symmetric. A general structure of the Slit kernel is shown in Fig. 1. At the smallest scale, the center slit of Slit$_\text{real}$ is made as narrow as allowed by the minimum spatial resolution of one pixel. The sum of the response for the whole receptive field is made zero, so there is no response when the entire field is exposed to the same light. Similar to Slit$_\text{real}$, the average response of Slit$_\text{imag}$ is also zero.

Different geometry for the regions has been tested. The proposed configuration, with the excitatory and inhibitory regions of Slit$_\text{imag}$ having the exact same shape as the excitatory region of Slit$_\text{real}$, shows the most consistent performance for the tests that we have performed. In addition, having the two kernels share certain geometric features further reduces computational requirements.

In constructing the real part of the Slit kernel, we seek to have its receptive field covering an area similar to its Gabor counterpart. We can use the Gaussian term in Eq. (1), $\exp\left[-(1/2)(x^2/\sigma_x^2 + y^2/\sigma_y^2)\right]$, as an estimation of the receptive field area. We assume that the receptive field is

approximated by the ellipse defined by the half-peak magnitude of the Gaussian term. A fixed scaling factor is used, such that $\sigma_{x0} = \sigma_{y0} = \sigma_\text{min} = 2.25$. Therefore, the area of the receptive field for the smallest Slit kernel $A_\text{RF}^{(0)}$ is estimated to be $A_\text{RF}^{(0)} = 22 \simeq 5^2$.

We set the smallest Slit$_\text{real}$ kernel to be a square of size 5×5. The center slit is one pixel wide and runs the full length of the receptive area. We call this 2-D function the mother kernel Slit$_\text{real}^{(0,0)}$, as all other real kernels are derived from it. The magnitude of its excitatory region is set to unity, while the magnitude of the inhibitory region is set to a negative value such that the kernel has an overall sum of zero. This kernel is depicted by the following equation, and in Fig. 2.

$$\text{Slit}_\text{real}^{(0,0)} = -\frac{1}{4} \cdot \text{rect}\left(\frac{x}{5}, \frac{y}{5}\right) + \frac{5}{4} \cdot \text{rect}\left(x, \frac{y}{5}\right), \qquad (2)$$

where

$$\text{rect}(x, y) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2} \text{ and } |y| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}. \qquad (3)$$

$\text{rect}(x, y)$ is a 2-D rectangle function with unit length and unit magnitude, centered at $(0,0)$.

The imaginary part of the kernel pair Slit$_\text{imag}$ is formulated similarly. It has the same receptive field as Slit$_\text{real}$. The excitatory region has the same shape and size as that of Slit$_\text{real}$, but is shifted. Its magnitude is also the same. An inhibitory region the same size as the excitatory region is located on the opposite side of the center axis. It has the same magnitude as the excitatory region, but with a negative sign. The mother kernel Slit$_\text{imag}^{(0,0)}$ is defined by the following equation and is shown in Fig. 3.

$$\text{Slit}_\text{imag}^{(0,0)} = \text{rect}\left(x - \frac{1}{2}, \frac{y}{5}\right) - \text{rect}\left(x + \frac{1}{2}, \frac{y}{5}\right). \qquad (4)$$

3.1.1 *Scaling of the Slit kernel*

When scaling the Slit kernel to the next larger size, all dimensions are increased by a factor of a but rounded to the nearest integer. At the same time, magnitudes of the responses are made inversely proportional to the area of its area. Because
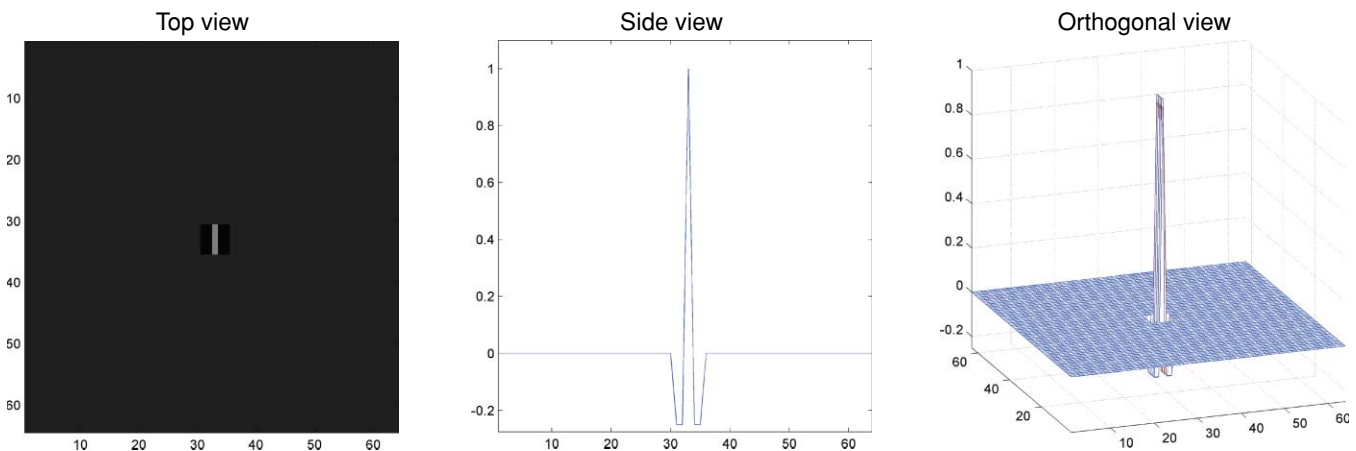
Top view   Side view   Orthogonal view

Fig. 2 A Slit$_\text{real}$ kernel at the smallest size with a receptive field of 5×5 pixels, shifted to the center of a 64×64 window.
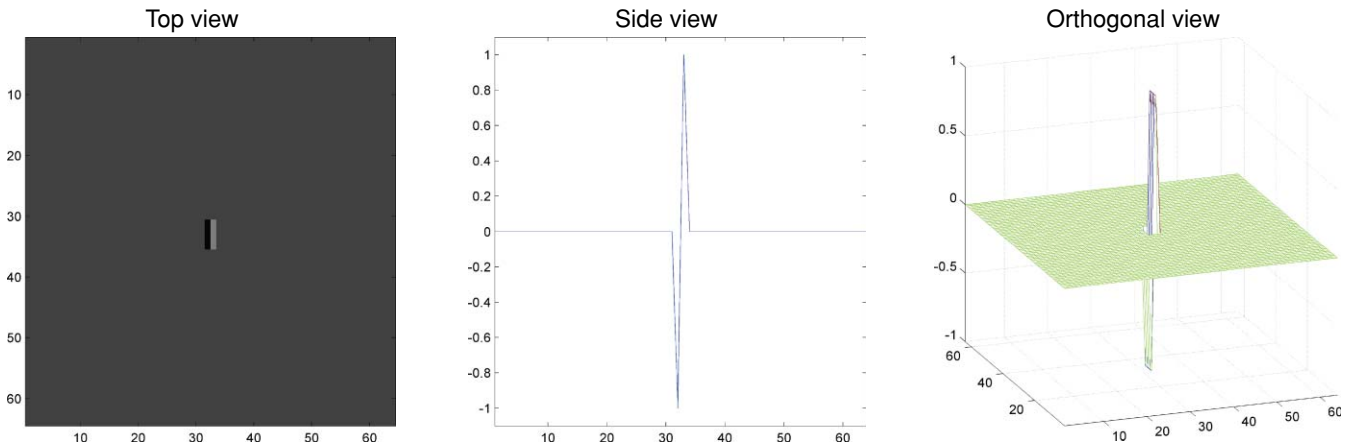
Fig. 3 A Slit$_{imag}$ at the smallest size with a receptive field of 5×5 pixels, shifted to the center of a 64×64 window.

of the discrete nature of pixels, ratios between areas of the excitatory region, inhibitory region, and receptive field can change during scaling. To compensate for these rounding effects, we use actual areas of the excitatory and inhibitory regions of the scaled kernel A_{Ex} and A_{In}, respectively, to calculate the actual magnitudes. The magnitude of the excitatory region M_{Ex} is first computed as follows:

$$M_{Ex}(\text{scaled}) = \frac{M_{Ex}(\text{original}) \cdot A_{Ex}(\text{original})}{A_{Ex}(\text{scaled})}. \quad (5)$$

Magnitude of the corresponding inhibitory region M_{In}, is set to a value that makes the overall filter have a zero mean.

$$M_{In}(\text{scaled}) = \frac{A_{Ex}(\text{scaled})}{A_{In}(\text{scaled})} \times M_{Ex}(\text{scaled}). \quad (6)$$

3.1.2 *Rotation of the Slit kernel*

For cases where the total number of orientations K is four, we need to generate Slit kernels at 0, 45, 90, and 135 deg. The cases for 0 and 90 deg are trivial. For the other two angles, we use rotated rectangles to represent the various regions.

Consider the Slit$_{real}$ kernel shown in Fig. 4. It has a receptive field defined by a 6×6 square, and an excitatory region of width two. This means the area of the receptive field is 36,
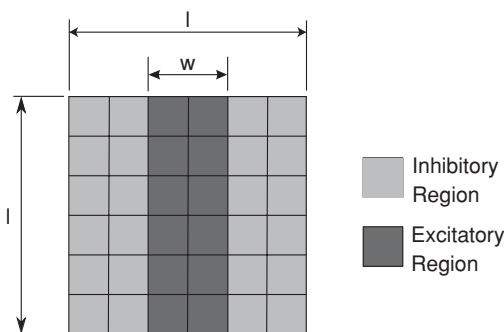
and those of the excitatory and inhibitory regions are 12 and 24, respectively. First, we need to determine the correct size of the receptive field for the rotated kernel. It can be shown that the area of a rotated square with r number of rows from the top to the widest point is given by the sequence {1, 5, 13, 25, 41, 61, ... }. We select the area in the list that is the closest to the area of the Slit kernel being rotated. For our example, the area of 41 at $r = 5$ is the closest match to the original area of 36. The next step is to add the two rows of excitatory regions. We start with the row in the center, and then move toward the right, and then left, until the desired number of rows is reached. The angle of these lines should follow the orientation of the rotated kernel. A Slit$_{real}$ that is rotated 135 deg is shown in Fig. 5.

To determine the magnitudes of the excitatory and inhibitory regions Eqs. (5) and (6) can be used.

The rotation of Slit$_{imag}$ can be performed in similar ways. The size of the receptive field is determined using the same method as for Slit$_{real}$. For Slit$_{imag}$, the excitatory and inhibitory regions always have the same number of columns. Therefore, they are drawn symmetrically on the two sides of the center line. To ensure symmetry, the center row is left as a no response region. Figure 6 shows a rotated Slit$_{imag}$
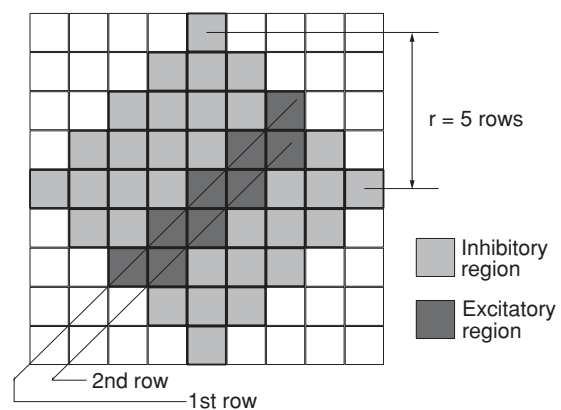

Fig. 4 An example of Slit$_{real}$ kernel at orientation 0 deg, with l = 6 and w = 2.

Fig. 5 A Slit$_{real}$ at 135-deg orientation, with two lines of excitatory region.
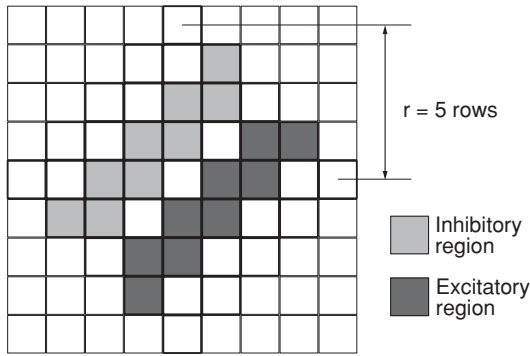
Fig. 6 A Slit$_{imag}$ at 135-deg orientation, with two lines of excitatory region and two lines of inhibitory region.

● Centroid of mini excitatory region

Fig. 8 Details of constructing rotated Slit kernel by shifting two halves.

kernel with the excitatory and inhibitory regions each two rows wide.

In cases where $K = 8$ is desired, we need to generate kernels that are 22.5 deg apart in orientation. The approximations shown in Fig. 7 can be used. Other angles of rotation can be handled in similar fashions.

3.1.3 *Feature representation using Slit kernels*

Generating the Slit feature for an image is similar to that for its Gabor feature. Let $\Xi_{0,0}$ be the mother Slit kernel, which is formed by combining Slit$_{real}$ and Slit$_{imag}$ with the smallest receptive fields, and at 0-deg orientation. That is, $\Xi_{0,0} = \text{Slit}_{real}^{(0,0)} + i \cdot \text{Slit}_{imag}^{(0,0)}$. Given an image $i(x, y)$, its filtering by a Slit kernel is defined as follows:

$$J_{m,n}(x, y) = \sum_{x_1} \sum_{y_1} i(x_1, y_1) \Xi_{m,n}(x - x_1)(y - y_1), \quad (7)$$

where $\Xi_{m,n}$ is a scaled and rotated Slit kernel. The mean $\mu_{m,n}$ and standard deviation $\sigma_{m,n}$ of the filtered terms are used to construct the Slit feature $\bar{f} = \{\mu_{0,0}, \sigma_{0,0}, \mu_{0,1}, \sigma_{0,1}, \ldots, \mu_{S-1,K-1}, \sigma_{S-1,K-1}\}$.

3.1.4 *Frequency response, coverage, and completeness*

The Slit kernels are constructed by 2-D rectangular regions that form an approximation of the Gabor function. Therefore, their frequency responses are combinations of the 2-D sinc pulses. Figures 9 and 10 shows the frequency responses of the
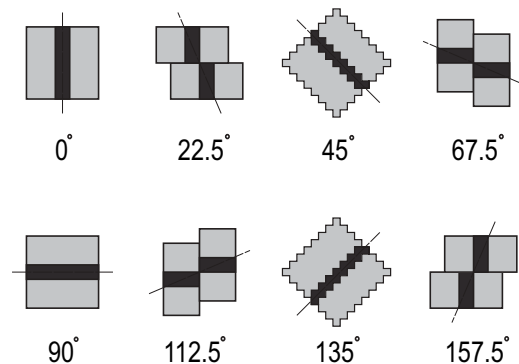

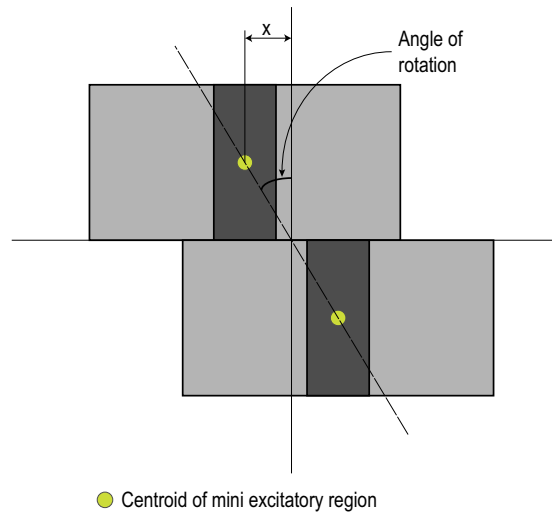**Fig. 7** The rotation sequence for Slit$_{real}$ when $K = 8$, where each kernel is 22.5 deg apart.

Slit and Gabor kernels at different scales. The Slit responses have more ripples but nevertheless have similar shapes and coverages as their Gabor counterparts. It is reasonable to argue that the Slit kernels, like the Gabor, are also complete but nonorthogonal. Techniques that are useful for reducing redundancy in Gabor analysis should also be applicable to Slit kernels.

3.1.5 *Fast algorithm for extracting Slit feature*

Instead of using FFT and IFFT, our algorithm performs convolution in the spatial domain. A fast algorithm based on integral images[11,14,15] is employed to perform filtering by Slit kernels.

Consider Fig. 11, where a Slit$_{real}$ and a Slit$_{imag}$ kernel at 0- and 135-deg are used to filter an image. Kernels at all scales and orientations can be decomposed into one of these forms. To calculate the filter response at these pixel locations, the kernels are pixel-wise multiplied with the corresponding image pixel values, and the results are summed. For each kernel, the desired result can be obtained by summing up the pixel values covered by two rectangles, and multiplying the resulting sums with magnitude of the respective rectangles. For Slit$_{real}$, the two rectangles are defined by the receptive field and the excitatory region, whereas for Slit$_{imag}$, they are the excitatory and inhibitory regions, as shown in Fig. 12.

Summing of the image pixel values over a rectangular region can be performed efficiently by using the idea of an integral image or summed area table (SAT).[14] Results for kernels with rectangles at 45 or 135 deg can be computed efficiently using the rotated summed area table (RSAT).[15]

The value of SAT(x, y) is defined as the sum of pixels inside the rectangle, ranging from the top-left corner at $(0, 0)$ to the bottom-right corner at (x, y), inclusively. With this table, the pixel sum RecSum(x, y, r) of any upright rectangle $r : (w, h, 0 \text{ deg})$ at location (x, y) can be determined using three additions. We can multiply RecSum(x, y, r) with its associated magnitude, and store the results inside a table called RecTable. Note that Rect$_1$ of Slit$_{real}$ and Rect$_1$ and Rect$_2$ of Slit$_{imag}$ share the same geometrical shape.
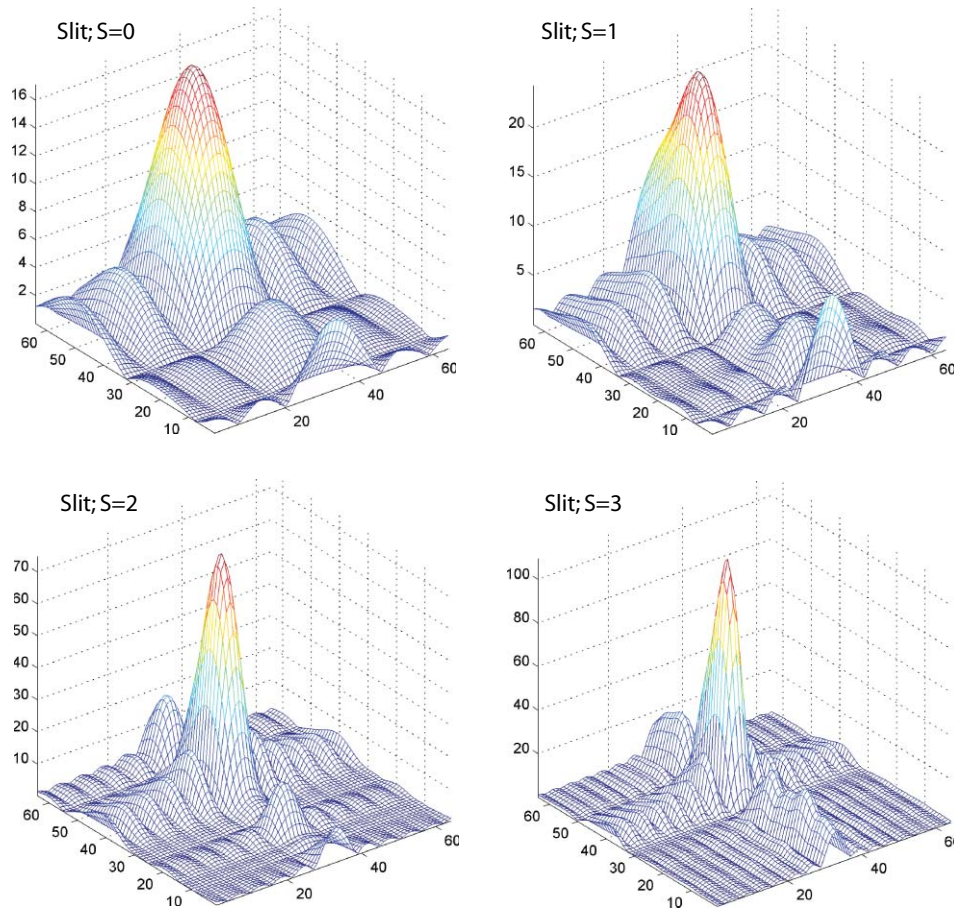
Fig. 9 Frequency responses of Slit kernels at different scales ($\Xi_{0,0}$ to $\Xi_{3,0}$, with $a = \sqrt{2}$).

Therefore, we can speed up the computation by reusing results during calculation of these RecTables.

Similarly, the value of RSAT(x, y) is defined as the sum of pixels inside a 90-deg wedge to the left of (x, y). Through the use of RSAT, the pixel sum RecSum(x, y, r) of a rotated rectangle r: (w, h, 45 deg) at location (x, y) can be determined by three additions. We can multiply RecSum(x, y, r) with its associated magnitude and store the results inside a RecTable.

Consider the areas to be summed. Both Rect$_1$ and Rect$_2$ of Slit$_{imag}$ have the same geometry and magnitude except for a sign change. Rect$_1$ of Slit$_{real}$ also shares the same geometry, but the magnitude is different. Two RecTables can be constructed to look up pixel sums for these three regions, with the correct magnitude already incorporated. For Rect$_2$ of Slit$_{real}$, a separate set of RecTable needs to be constructed. In total, two sets of RecSum and three RecTables are required to provide sum-of-pixel values with the correct magnitudes incorporated to all rectangular regions involved.

Filtering by kernels with odd angle rotations (e.g., 30 deg) can also utilize the fast algorithm. All the regions are rectangles, so SAT can be used to reduce computation complexity. Since all the rectangles are of two sizes only, the same two sets of RecSum and three RecTables can be used to assist with feature extraction.

We now compare the computational requirements between the Slit feature using the fast algorithm and the Gabor feature using FFT plus IFFT. It is assumed that the feature

for the whole image is desired and the image size is N^2, with N being a power of 2, so FFT can be applied for the case of the Gabor kernel. With our proposed Slit kernel and the fast algorithm, features at each pixel position can be extracted individually instead of having to calculate for the whole image. For comparison purposes, only additions and multiplications are counted, but not table look-ups or indexing.

Table 1 shows a summary of the computational requirements for each of the steps. We also show the actual numbers of numerical operations in Table 2, assuming the image size is 64×64. Savings in computation are substantial. Filtering with Slit kernels using the fast algorithm requires only 11 to 17% of the number of operations when compared to filtering with Gabor kernels using FFT. Savings would be even greater for larger images. Savings for kernels with odd rotation angles will be slightly less, as the convolution with these kernels needs an additional $4N^2$ floating point additions during the calculation of the sum of pixels for the added rectangular regions.

3.2 *Circular Sum of Feature Elements*

We now describe an enhancement to the Gabor or Slit feature to make it more rotation invariant. Consider an image i with a Gabor or Slit feature $\bar{f}^{(i)}$, where

$$\bar{f}^{(i)} = \{\mu_{0,0}, \sigma_{0,0}, \mu_{0,1}, \sigma_{0,1}, \ldots, \mu_{S-1,K-1}, \sigma_{S-1,K-1}\}. \quad (8)$$
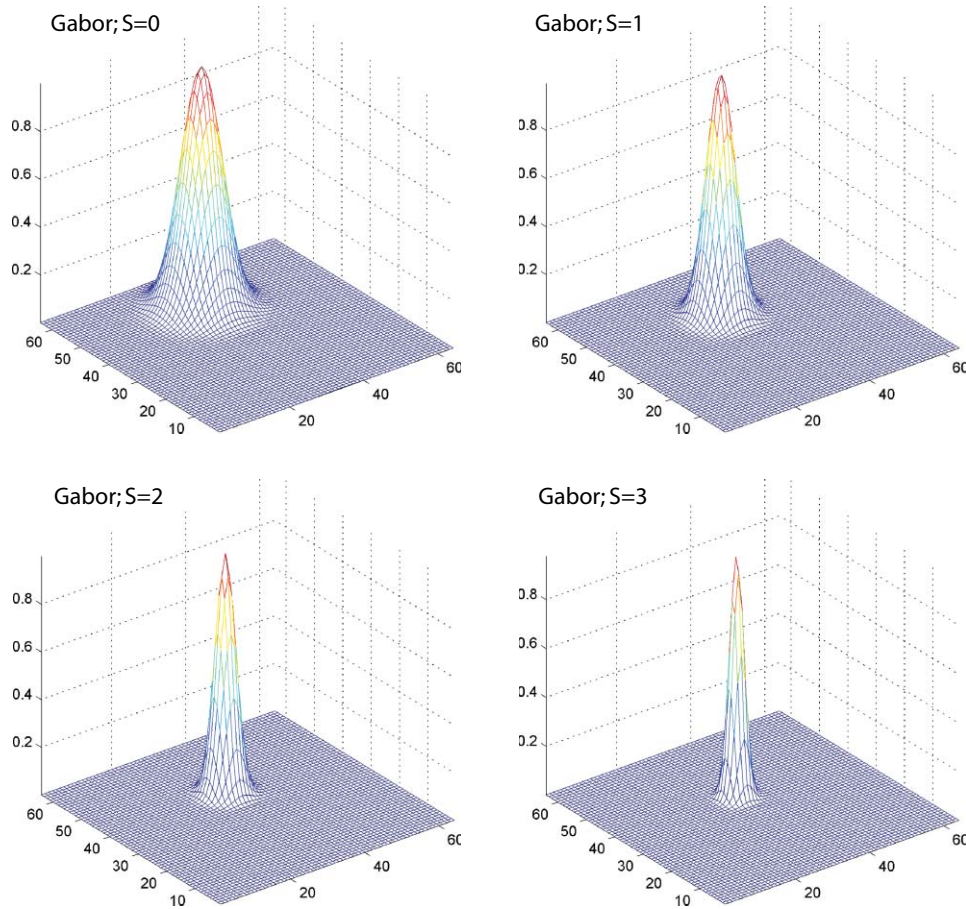
Fig. 10 Frequency responses of Gabor kernels at different scales ($g_{0,0}$ to $g_{3,0}$, with $a = \sqrt{2}$).

$\mu_{m,n}$ and $\sigma_{m,n}$ are the mean and standard deviation of the filtered output by the kernel at the m'th scale and n'th orientation. Both m and n are integers, and $0 \leq m < S - 1, 0 \leq n < K - 1$. The length of this feature is $2 \times S \times K$.

Consider an image set with a mixture of rotated texture images that are visually similar. There are certain applications where it is desirable to classify these visually

similar images as belonging to the same group. Both the Gabor and Slit kernels have fairly high directional selectivity. Therefore, if the feature as defined in Eq. (8) is used directly for the calculation of similarity, some rotated images will be considered quite different from each other, and will be classified incorrectly. To improve the rotation invariance of the
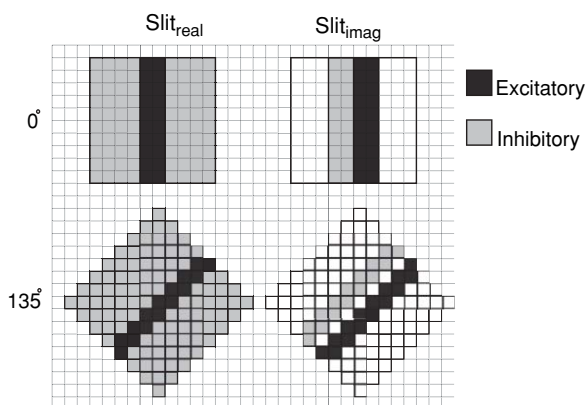

Fig. 11 The four fundamental configurations of filtering with Slit kernels. Each of the configurations has exactly two homogeneous regions to be summed.
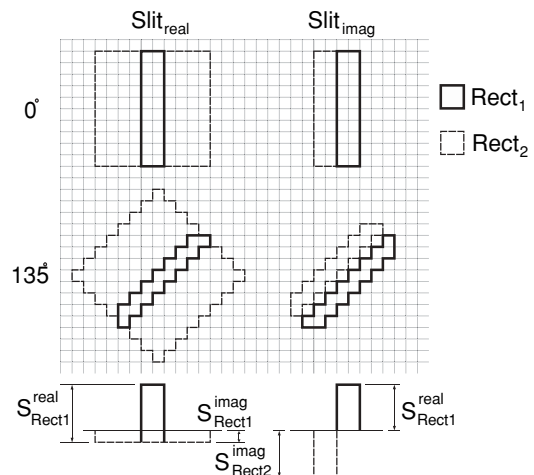

Fig. 12 Definition of rectangular regions and their magnitudes for Slit$_{\text{real}}$ and Slit$_{\text{imag}}$.

Table 1 Comparison of computation complexities between filtering with Gabor and Slit kernels. Note that images are sized $N \times N$ pixels, where N is a power of 2, so FFT can be used for Gabor filtering.

	Steps	Float-point		Integer
		$+$	$\times$	$+$
Gabor	Compute FFT of image	$3N^2 \log_2 N^2$	$2N^2 \log_2 N^2$	
	$FFT_{\text{image}} \times FFT_{\text{Gabor}}$	$2N^2$	$4N^2$	
	Compute IFFT	$3N^2 \log_2 N^2$	$2N^2 \log_2 N^2$	
	Total for first	$6N^2 \log_2 N^2 + 2N^2$	$4N^2 \log_2 N^2 + 4N^2$	
	Total for subsequent	$3N^2 \log_2 N^2 + 2N^2$	$2N^2 \log_2 N^2 + 4N^2$	
Slit - using SAT	Compute SAT			$3N^2$
	Compute two sets of RecSum			$6N^2$
	Multiply RecSum with magnitudes to obtain three RecTables		$3N^2$	
	Sum results from RecTables to form result [real and imaginary]	$2N^2$		
	Total for first	$2N^2$	$3N^2$	$9N^2$
	Total for subsequent	$2N^2$	$3N^2$	$6N^2$
Slit - using RSAT	Compute RSAT			$5N^2$
	Compute two sets of RecSum			$6N^2$
	Multiply RecSum with magnitudes to obtain three RecTables		$3N^2$	
	Sum results from RecTables to form result [real and imaginary]	$2N^2$		
	Total for first	$2N^2$	$3N^2$	$11N^2$
	Total for subsequent	$2N^2$	$3N^2$	$6N^2$

feature, we propose to sum the elements in the feature vector that belong to the same scale. This will reduce directionality of the feature.

The circularly summed feature of an image i is defined as follows:

$$\bar{f}_{cs}^{(i)} = \{\mu_0, \sigma_0, \mu_1, \sigma_1, \ldots, \mu_{S-1}, \sigma_{S-1}\}, \tag{9}$$

where

$$\mu_m = \sum_n \mu_{m,n}, \quad \sigma_m = \sum_n \sigma_{m,n}. \tag{10}$$

The resulting feature is only $2 \times S$ in length, which means the feature space is significantly lower in dimension. This

Table 2 Number of numerical operations for image filtering using Gabor and Slit kernels when image size is 64×64. Note that percentage calculations for *first* and *subsequent* are with respect to their respective Gabor values.

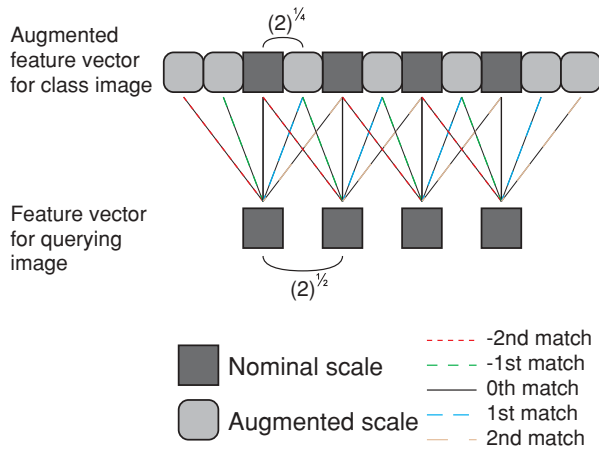|			Floating point		Integer	Total	% of Gabor
			$+$	$\times$	$+$		
Gabor		First	303,104	212,992		516,096	100%
		Subsequent	155,648	114,688		270,336	100%
Slit	Using SAT	First	8192	12,288	36,864	57,344	11.1%
		Subsequent	8192	12,288	24,576	45,056	16.7%
	Using RSAT	First	8192	12,288	45,056	65,536	12.7%
		Subsequent	8192	12,288	24,576	45,056	16.7%

Fig. 13 Different scaling factors are used for class images and for images to be classified. In this example, $b = 2$ and $a = \sqrt{2^{1/2}} = \sqrt{2^{1/4}}$. Each node denotes two elements, a sum of means and a sum of standard deviations.

Fig. 14 Sliding of the feature vector for a querying image along the augmented feature vector of a class image, searching for the best match. In this example, $S = 4$, $a_{\text{image}} = \sqrt{2}$, and $b = 2$. Number of matches is five.

can have positive effects on classification performance and computational efficiency.

3.3 Scale Invariance by Slide Matching

We now develop an enhancement to make a feature more scale invariant. Assume that a group of images are to be classified into a number of classes, with each class represented by a class image. The images to be classified are filtered using a set of kernels with S scales and K orientations. A circular-sum feature is generated for each image. A value of $\sqrt{2}$ is used as the nominal scaling factor a_{image}. The resulting feature has a length of $2 \times S$.

For the class images, a different set of kernels is used. These kernels have a smaller scaling factor a_{class}. The goal is to generate additional feature elements at several augmented scales that are in between the nominal scales. This will allow multiple distance measurements around the nominal scale to be calculated. The best match, or the shortest distance, around the nominal scale is used as the final distance. Figure 13 illustrates the relationship between the nominal and augmented elements.

For the class images, let b be the number of augmented scales per nominal scale used for feature extraction. In this work, we choose $b = 2$, although different numbers can be used. To space the augmented scales evenly between the nominal scales, the scaling factor for class images a_{class} is determined as follows:

$$a_{\text{class}} = (a_{\text{image}})^{\frac{1}{b}}. \tag{11}$$

The resulting augmented feature vectors for the class images are of lengths $2[2S + (2b - 1)]$.

During the classification of a querying image, its feature vector is matched to a subset of a class feature vector at different offsets, as illustrated in Fig. 14. For a given b, the number of matches to be performed is $(2b + 1)$. At each matching position, the distance between the query image and the class image is calculated. The smallest distance from all of these matching locations is chosen as the distance between the image and this class. This process is repeated for all of the classes. The image is said to belong to class C_i if the
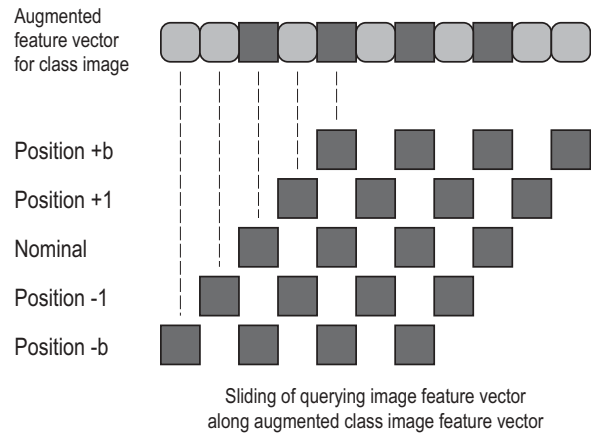
distance between the query image and the class image of C_i is the smallest among all of the classes.

3.4 Distance Measure

Based on the Gabor or Slit feature, the difference between two images D_f can be computed using any kernel-based distance measure. To improve the retrieval performance, differences between the gray level intensities D_s are also used. These two distance measures are combined as follows to form the actual distance:

$$D = \frac{D_f}{\sigma_f} + \frac{D_s}{\sigma_s}, \tag{12}$$

where σ_f and σ_s are the standard deviations of the respective distance measures obtained during the training phase.

4 Experimental Results

To examine the effectiveness of the proposed enhancements outlined in the previous sections, we have performed extensive experiments using different test images. We compare the retrieval performances based on the Slit and Gabor features, and examine whether the Slit feature's discriminating power on texture images is comparable to that of the Gabor feature. We also compare our proposed algorithm with published results.

4.1 Comparing Performance between Slit and Gabor Features

We use all 111 images from the Brodatz database[16,17] in the experiments. Three groups of texture images are created. The first group is called "divided," where for each original image we extract 16 subimages, each 128×128 in size, from a fixed set of offsets in a 4×4 grid. The second group, called "rotated," is created by first rotating the original image by an angle of $n\pi/16$, where $n = 0, 1, \ldots, 15$. At each orientation, the center region of size 128×128 pixels is extracted to form the subimage. The last group, called "scaled," is generated by first cropping the original image to a size of $(640 - 30m) \times (640 - 30m)$ pixels in size, where $m = 0, 1, 2, \ldots, 15$. These cropped images are then scaled down to a size of
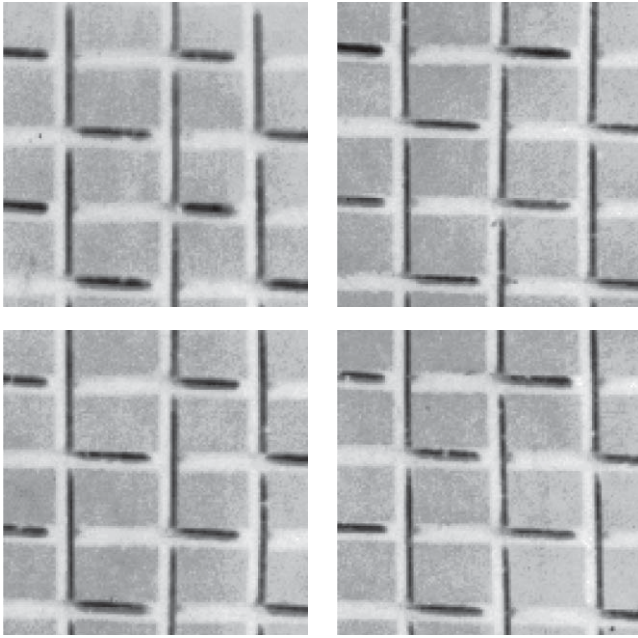
Fig. 15 Example images from the group "divided."

Fig. 17 Example images from the group "scaled."

128×128 pixels using bicubic interpolation. Each of these groups contains a total of $111\times16 = 1776$ images. Some example images from each of these groups are shown in Figs. 15, 16, and 17.

For each group, we extract both the Slit and the Gabor features from each of the images. All 1776 images in each set are used in turn to retrieve their closet matches from the same group using either the Slit or Gabor feature for distance measure. The L_2-norm is used without applying the circular-sum or slide-matching enhancements. There are a total of 16 ground-truth images for each query. Recall, defined as
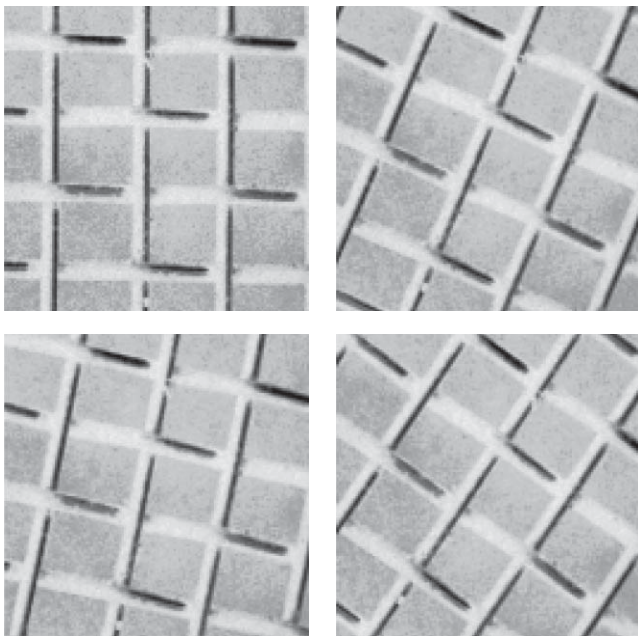
follows, is used to compare the performance between the two features.

$$\text{recall} = \frac{\text{number of correctly retrieved images}}{\text{number of retrieved images}}. \quad (13)$$

We evaluate and compare the respective retrieval performances of the two features using different number of scales S and orientations K. In the literature,[5,12,18–20] $S = 3, 4, 5, 6, 8$ and $K = 4, 5, 6, 8$ have been reported as effective. In the experiments, we set $K = 4, 6,$ and 8 only, since the Slit kernels are only defined for these orientations. S is set at 5 only.

Figure 18 shows the recall rates of the two features for $K = 4, 6,$ and 8, respectively, for the divided set. In general, the performances of the two features are very similar. Figure 19 shows the corresponding recall rates when the rotated set is used. Generally speaking, the retrieval performance for the rotated images is not as accurate as that for the divided image set. This is because both the Gabor and Slit kernels are quite directional, so their associated features are sensitive to variations in orientation. In general, for all K, the retrieval performance of the Slit feature exceeds that of the Gabor feature. This is likely because the Slit kernels are slightly less directional than the Gabor ones. The performance of the two features for the scaled set is shown in Fig. 20. The overall retrieval accuracies of the two features based on the scaled set are better than for the rotated set, but are worse than for the divided set. With the scaled set, the Slit feature outperforms the Gabor feature in all cases.

Experimental results in this section indicate that the Slit feature is useful for the retrieval of texture images. The performance of the Slit feature has been shown to be similar to that of the Gabor feature, and sometimes slightly better. The performance for features with different numbers of scales and orientations is also quite consistent.
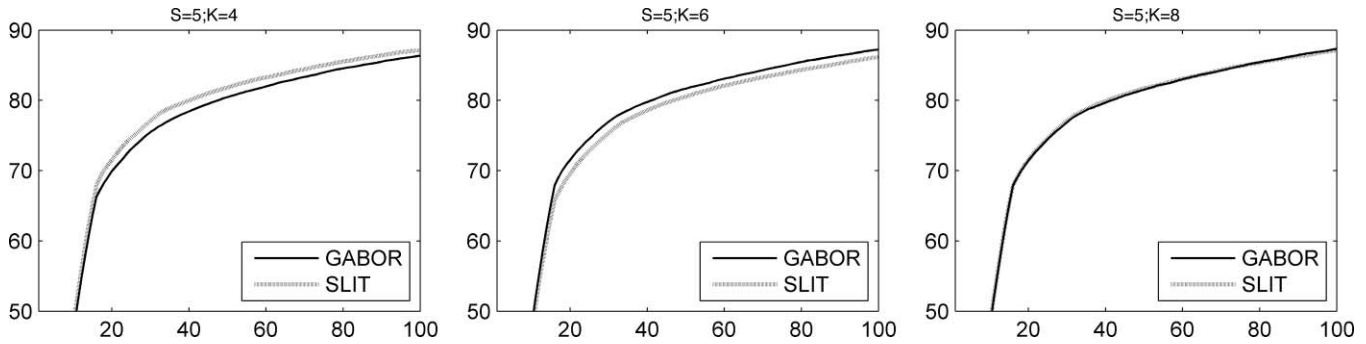

Fig. 16 Example images from the group "rotated."

Fig. 18 The recall rates of the Slit and Gabor features based on the divided set. The horizontal axes are the number of retrieved images, while the vertical axes are the recall rates in percentage.

Fig. 19 The recall rates of the Slit and Gabor features based on the rotated set. The horizontal axes are the number of retrieved images, while the vertical axes are the recall values in percentage.
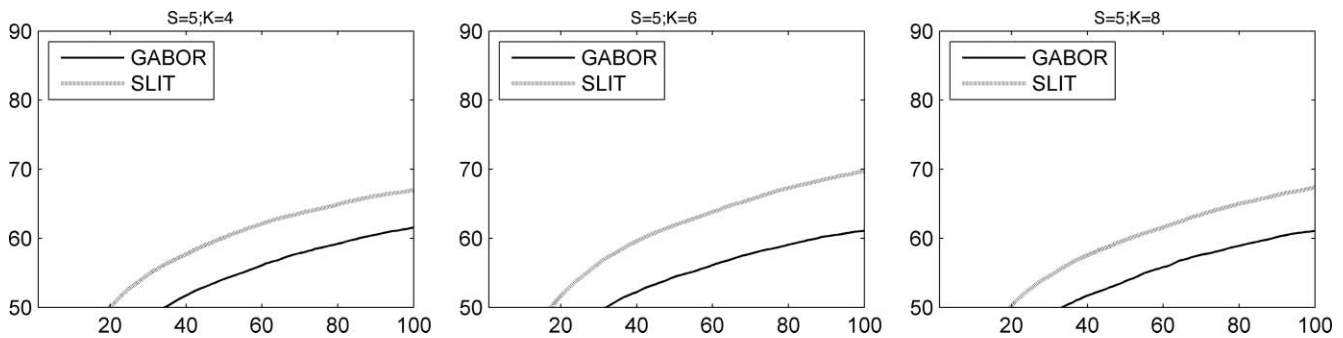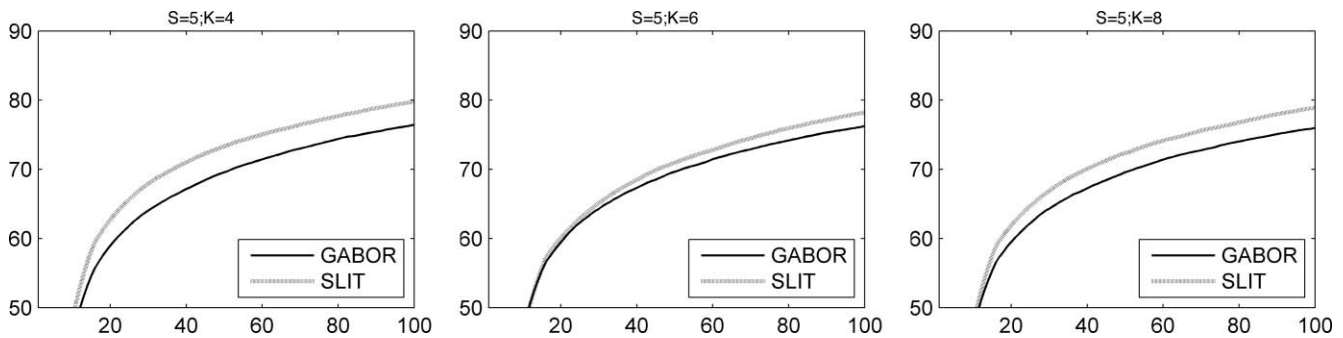

Fig. 20 The recall rates of the Slit and Gabor features based on the scaled set. The horizontal axes are the number of retrieved images, while the vertical axes are the recall values in percentage.
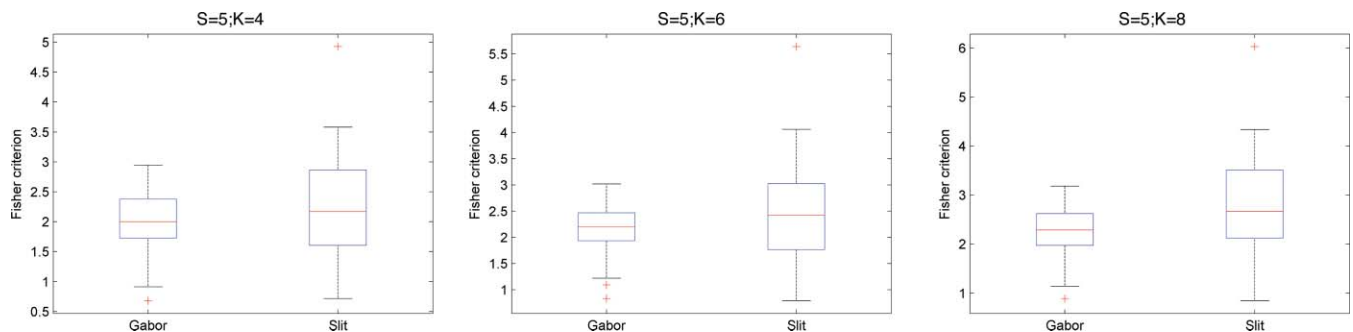

Fig. 21 Box and whisker plots of the values of f for the Gabor and Slit vectors. Nine Brodatz images are used.
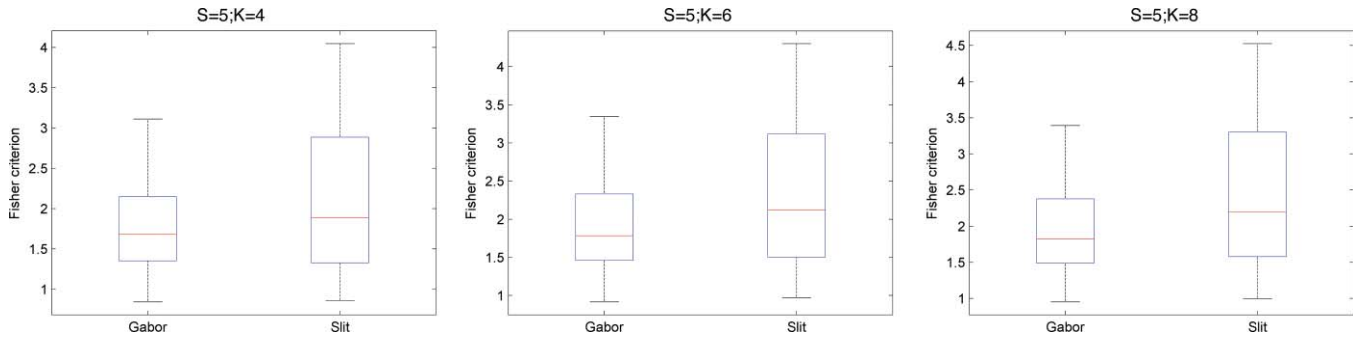
Fig. 22 Box and whisker plots of the values of *f* for the Gabor and Slit vectors. Nine Outex images are used.

4.2 Comparing Slit and Gabor Kernels Using Fisher Criterion

In this section, we employ a different method to compare the discriminating power of the two kernels. References 18, 21, and 22 use the Fisher criterion[23,24] to compare the performance of different features. This approach is statistically based and can be used to compare the effectiveness of the respective features, regardless of the subsequent classifiers being used. It provides a quantitative measure of the separability between clusters of feature elements, as projected onto the best 1-D axis.

The projection of a feature *x* onto the 1-D space is defined as follows:

$$y = (u_1 - u_2)^T \sum_p^{-1} x, \qquad (14)$$

where u_1 and u_2 are the means of the two clusters, and $\sum_p$ is the pooled covariance matrix. Using this projection results in the following ratio being maximized.

$$f = \frac{|\eta_1 - \eta_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}}, \qquad (15)$$

where σ_1 and σ_2 are the variances of the clusters of the projected feature, and η_1 and η_2 are their corresponding means. This ratio *f*, known as the Fisher criterion, expresses the separation between two clusters relative to the sum of their variances, or spreads. A larger value indicates a better separation.

We start with a setup based on Ref. 22 using the same nine images from the Brodatz set. We crop each image to a size of 128×128 pixels, and then use one set of Gabor kernels and one set of Slit kernels, each with *S* scales and *K* orientations, to extract features from the image. At each pixel position, a feature vector of length $S \times K$ is extracted. A total of $128 \times 128 = 16,384$ feature vectors are extracted for each image.

For each pair of images, the pooled covariance matrix $\sum_p$ and the vector means u_1 and u_2 of their corresponding vectors are calculated. The two clusters of vectors are projected onto the 1-D space using Eq. (14). In the projected space, the Fisher criterion as defined in Eq. (15) is evaluated. In total, there are $8 + 7 + \cdots 1 = 36$ image pairs.

Box and whisker plots of the values of *f* based on the Gabor and Slit kernel sets at different *S* and *K* values are examined to compare the discriminating properties of the two kernels. The distributions of the values *f* for the Slit kernels and the Gabor kernels are quite similar. Except

for $S = 3$, $K = 4$, and $S = 7$, $K = 4$, the medians of the value *f* for the Slit vectors are roughly the same or higher than those for the Gabor vectors. This indicates that the Slit vectors have slightly higher discriminating power between these nine images. For most cases, the values of *f* for the Slit vectors tend to have wider spreads. Several samples of the plots are shown in Fig. 21.

The same experiment is also conducted by using part of the Outex set, as well as the full sets of the Brodatz and Outex databases. Nine images from the Outex set,[25] namely barleyrice008, chips016, flour003, paper006, pasta004, plastic021, seeds012, wallpaper001, and wool002, are used. With the Outex set, the medians of the values of *f* for the Slit vectors are somewhat higher than those for the Gabor vectors on the right, except for $S = 3$, $K = 4$, and $S = 7$, $K = 4$. In addition, the spreads for both vectors are larger for these images. Figure 22 illustrates some box and whisker plots of the *f* values. We have also calculated the *f* values of the whole Brodatz and Outex image sets at $S = 4$ and $K = 6$. The box and whisker plots are shown in Fig. 23. Again, the median value and the upper quartile for the Slit feature are slightly higher than those for the Gabor feature. Furthermore, the values for the Slit feature have a lot more outliers toward the high end, indicating that the Slit kernels produce some highly separable clusters of features.

4.3 Classifying Rotated Images with Circular Sum of Features

In this section, we evaluate the effectiveness of the circular sum of features. We adopt the texture classification experiment as outlined in Ref. 4. Two sets of images, A.I and A.II, are used, both derived from the Brodatz database. The circular sum is applied to both the Gabor and Slit features of all images. We use $S = 5$, $K = 4$, and $a = \sqrt{2}$ for this experiment. Classification results for set A.I using the circularly summed Gabor and Slit features, together with those reported in Ref. 4, are listed in Table 3. For this small image set, both the Slit and Gabor features with the circular sum perform well.

The classification results for set A.II are listed in Table 4. The performance of the Slit feature with the circular sum is similar to that of the Gabor feature with a dominant orientation, while the Gabor feature with the circular sum performs the best. However, the computation of the Slit feature involves 20 Slit kernels only, whereas the Gabor features in Ref. 4 use 30 Gabor kernels. Each filtered output for a Slit kernel requires 11 to 17% of computations
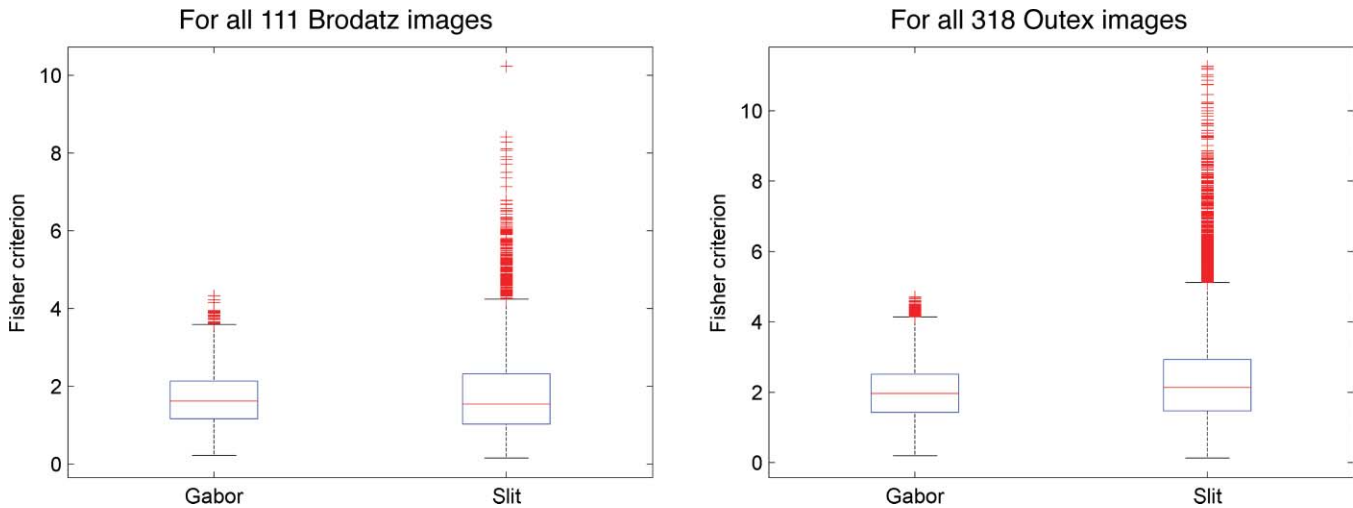
Fig. 23 Box and whisker plots of the values of *f* for the Gabor and Slit vectors, at $S = 4$ and $K = 6$.

when compared to Gabor filtering. Therefore, the overall computation requirements for the Slit feature in this experiment are only 7 to 11% of those for the Gabor feature. Therefore, if speed is important, we can use the Slit feature with the circular sum to achieve a good performance at low computational costs. On the other hand, if the ultimate performance is desired, we can use the Gabor feature with circular sum for rotation-invariant classification.

4.4 Image Classification with Circular Sum and Slide Matching

To test the effectiveness of circular sum plus slide matching on classifying images under rotation and scaling, we compare classification performance with two published sets of results. In Ref. 6, Xie et al. apply enhancements to the Gabor features in obtaining some of the best published results, while in Ref. 26, Pun and Lee use an enhanced wavelet-based approach.

4.4.1 Comparing with adaptive circular orientation normalization and elastic cross-frequency searching

Xie et al.[6] propose using a combination of adaptive circular orientation normalization (ACON) and elastic cross-frequency searching (ECFS) to achieve rotation-and scale-invariant image classification. Images from both the Brodatz and Outex databases are tested.

Three sets of images are created using all 111 base images from the Brodatz database. All images are of sizes 64×64 pixels. The 111 images at 0-deg rotation and 1.0 scale are used as the class images. Images from the test sets are classified into these 111 classes based on our proposed algorithm, and the performance is compared to that listed in Ref. 6. A summary of the classification accuracies is shown in Table 5. For the results in Ref. 6, $S = 5$ and $K = 10$ are used. For the Gabor features with the circular sum, slide matching and the simple fusion of image features with statistical intensity values, $S = 5$, $K = 8$, $a = \sqrt{2}$, and $b = 2$ are used; while for the Slit feature, the corresponding settings are $S = 6$, $K = 4$, $a = \sqrt{2}$, and $b = 2$.

For the rotated images in set X.I, both the Gabor and Slit features with the enhancements proposed in this work perform better than Ref. 6. For the scaled images in set X.II, both perform worse than that in Ref. 6, while the Slit feature has the lowest accuracy. For the mixture of rotated and scaled images in set X.III, the Gabor feature with the proposed enhancements achieves the best performance, while the Slit feature performs the worst. The images in set X.III are more diverse, and should better reflect what we will see in practical applications.

With the Outex database, three sets of images are also created using all 318 image classes. A summary of the classification results is shown in Table 6, where $S = 5$ and

Table 3 Classification accuracies of rotated images set A.I for 91 images into 13 classes.

	Gabor feature with enhancement from Ref.	Gabor feature circularly summed	Slit feature circularly summed
Number of correctly classified images	87	89	91
Mean success rate	95.6%	97.8%	100%

Table 4 Classification accuracies of rotated images set A.II for 3996 images into 111 classes. Number of nodes is $S \times K$. Each node contains one μ and one σ.

Feature vectors with different scales (S) and orientations (K)	Number of nodes	Mean success rate
Gabor (F6) – $S = 5$; $K = 6$; dominant orientation, from Ref. 4	30	93.4%
Gabor – $S = 4$; $K = 6$; with circular-sum.	24	99.1%
Slit – $S = 5$; $K = 4$; with circular sum.	20	92.9%

Table 5 Classification accuracies of different image sets based on the Brodatz dataset. Intensity value in the first classification is a result from Ref. 6.

	Set X.I	Set X.II	Set X.III
GW + ACON + ECFS + intensity value	95.0%	87.1%	80.1%
Gabor + circular sum + slide matching + intensity	98.9%	82.5%	82.5%
Slit + circular sum + slide matching + intensity	98.4%	75.5%	75.6%

Table 6 Classification accuracies of different image sets based on the Outex dataset. Intensity value is a result from Ref. 6.

	Set O.I	Set O.II	Set O.III
Gabor + circular sum + slide matching + intensity	97.8%	97.2%	89.7%
Slit + circular sum + slide matching + intensity	98.5%	92.4%	82.7%
GW + ACON + ECFS + intensity value	92.9%	76.5%	72.8%

$K = 10$ are used for Ref. 6; $S = 5$, $K = 8$, $a = \sqrt{2}$, and $b = 2$ for the Gabor feature; and $S = 6$, $K = 4$, $a = \sqrt{2}$, and $b = 2$ for the Slit feature. The results show that the classification performance of both the Gabor and Slit features with our proposed enhancements is better than that reported in Ref. 6 by fairly wide margins. In general, Gabor features with our enhancements perform better than Slit features with our enhancements.

4.4.2 *Comparing with log-polar wavelet energy signatures*

In Ref. 26, Pun and Lee propose using a log-polar wavelet energy signature for rotation- and scale-invariant texture classification. Five image sets are used in the experiments; all of these sets are created from 25 base images chosen from the Brodatz set. A Mahalanobis classifier is used for distance measurement between images, and a feature length M of 96 is used. To have a fair comparison, we also adopt a Mahalanobis classifier in our algorithm. A modified form is used, because slide matching requires distance calculations with multiple augmented scales. We use $b = 2$ in this experiment, meaning five distances are calculated for each match. Features at all augmented scales are generated for the images, which are used to calculate the sample means and sample covariance matrices for each augmented scale. Each augmented scale is treated as a subclass within the class. During matching, the Mahalanobis distance is calculated between the query image and all five subclasses. The smallest value from these five distances is used as the distance measure with this class. The Mahalanobis distance between an image with feature x and a subclass i with mean u_i and covariance

matrix $\sum_i$ is defined as follows:

$$d(x, u_i) = (x - u_i)^t \sum_i (x - u_i). \quad (16)$$

In this comparison, the settings for the Gabor feature are $S = 5$, $K = 8$, $a = \sqrt{2}$, and $b = 2$, while those for the Slit feature are $S = 6$, $K = 4$, $a = \sqrt{2}$, and $b = 2$. The classification results of the log-polar wavelet energy signature and our proposed algorithms are summarized in Table 7. The performance of our proposed algorithms is better than that of the log-polar wavelet energy signature in all cases. We have also tested classification performance without using the statistical intensity information. Accuracy levels increase, likely because the Mahalanobis classifier is already very accurate. Fusing its output with a less precise distance measure based on intensity will result in a less accurate distance measure. Although the Mahalanobis classifier used in this comparison can be highly accurate, it may not be suitable for use in many applications. This classifier requires significantly more computational resources. Furthermore, all features from the image set are used for the calculation of the class means and covariance matrices.

4.5 *Computational Efficiency of Slit Features*

We have compared the theoretical computational times required for feature extraction using our fast algorithm and the traditional algorithm based on FFT and IFFT. Our analysis shows that extracting a Slit feature using the fast algorithm requires only 11 to 17% of the operations required for a Gabor feature. In this section, we seek to verify this result using a notebook computer (Core2 Duo 2.4 GHz; 4-GB RAM; Windows XP SP3) as the target platform.

Table 7 Classification accuracies of image sets based on Pun and Lee,[26] and the proposed algorithms. The log-polar wavelet energy signal is a result from Ref. 26.

	Set P.I	Set P.II	Set P.III	Set P.IV	Set P.V
Gabor + circular sum + slide matching + intensity	93.6%	95.4%	93.9%	100%	100%
Gabor + circular sum + slide matching	98.7%	100%	99.5%	100%	100%
Slit + circular sum + slide matching + intensity	93.4%	93.8%	94.5%	100%	100%
Slit + circular sum + slide matching	97.3%	100%	99.9%	100%	100%
Log-polar wavelet energy signature	90.8%	93.8%	88.6%	100%	82.5%

Table 8 Runtimes for 10,000 cycles of convolution using our fast algorithm, or FFT plus IFFT.

	MATLAB	C++ with no optimization	C++ optimized for size	C++ with full optimization
Fast algorithm (t_{fast})	8.5 sec	2.0 sec	0.94 sec	0.48 sec
FFT + IFFT (t_{FFT})	7.1 sec	5.4 sec	4.2 sec	4.1 sec
$t_{fast}/t_{FFT} \times 100\%$	120%	37%	22%	12%

We have implemented the core functions listed in Table 1 using Visual C++. The fast algorithm is implemented using "for" loops, while a highly optimized library called FFTW[27] is used for the FFT-based algorithm. First, we apply no optimization to the code compilation and linking; then we optimize for size; finally, full optimization is allowed. We perform the convolution operations using both our fast algorithm through SAT and the FFT-based algorithm, 10,000 times each. The resulting time lapses are summarized in Table 8. It is obvious that the time savings are highly dependent on the level of optimization, the capabilities of the compiler, and the underlying processor. In any case, by utilizing the maximum level of optimization, the fast algorithm does demonstrate the execution speed increase suggested by our analysis.

5 Conclusions, Discussions, and Future Work

The analysis of texture patterns can improve the performance of content-based image retrieval systems. Therefore, it is desirable to build a robust framework for the classification and retrieval of texture images. Being both repetitive and stochastic, the variable nature of texture presents a unique set of challenges to both the research community and practitioners.

To enhance the performance of multiresolution Gabor analysis on texture images, we have made a number of proposals. The first is a newly developed Slit kernel, which is inspired by experimental results for the visual cortex. The associated Slit feature can be computed efficiently using a fast algorithm. In applications where computational speed is important, this kernel can replace the Gabor kernel. The second proposal is a circular-sum operation for the feature elements belonging to the same scale. This enhancement is an attempt to improve rotation invariance. The third proposal is a scheme for similarity measurements, based on the slide matching of distances between the nominal and augmented scales. This has the potential of improving the classification of images, even when they are different in scale.

We present experimental results for the enhancements that we have developed. The Slit kernel is shown to have similar performance as the Gabor kernel in multiresolution analysis, while requiring significantly lower computation during feature extraction. The circular-sum operation on the feature vector is compared with an enhancement described in a recent publication, and it demonstrates a superior performance for the classification of rotated images. Finally, a combination of the proposed enhancements is used to classify rotated and scaled images from two large databases. The classification accuracies compare favorably with published results.

References

1. F. Mendoza, P. Dejmekb, and J. M. Aguilera, "Colour and image texture analysis in classification of commercial potato chips," *Food Res. Intl.* **40**(9), 1146–1154 (2007).
2. F. Lahajnar and S. Kovacic, "Rotation-invariant texture classification," *Patt. Recog. Lett.* **24**(9–10), 1151–1161 (2003).
3. T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," *Computer Vision*, pp. 404–420 (2000).
4. S. Arivazhagan, L. Ganesan, and S. P. Priyal, "Texture classification using gabor wavelets based rotation invariant features," *Patt. Recog. Lett.* **27**(16), 1976–1982 (2006).
5. J. Han and K. K. Ma, "Rotation-invariant and scale-invariant gabor features for texture image retrieval," *Image Vision Comput.* **25**(9), 1474–1481 (2007).
6. X. Xie, Q. Dai, K. M. Lam, and H. Zhao, "Efficient rotation- and scale-invariant texture classification method based on gabor wavelets," *J. Electron. Imag.* **17**(4), 043026 (2008).
7. V. Kyrki, J. K. Kämäräinen, and H. Kälviäinen, "Simple gabor feature space for invariant object recognition," *Patt. Recog. Lett.* **25**(3), 311–318 (2004).
8. J. Ilonen, J. K. Kämäräinen, and H. Kälviäinen, "Efficient computation of gabor features," Tech. Repo. Lappeenranta Univ. of Technology, Finland (2005).
9. C. C. Chen and D. C. Chen, "Multi-resolutional gabor filter in texture analysis," *Patt. Recog. Lett.* **17**(10), 1069–1076 (1996).
10. K. R. Namuduri, R. Mehrotra, and N. Ranganathan, "Efficient computation of gabor filter based multiresolution responses," *Patt. Recog.* **27**(7), 925–938 (1994).
11. W. P. Choi, S. H. Tse, K. W. Wong, and K. M. Lam, "Simplified gabor wavelets for human face recognition," *Patt. Recog.* **41**(3), 1186–1199 (2008).
12. B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Patt. Anal. Mach. Intel.* **18**(8), 837–842 (1996).
13. C. K. Chui, *An Introduction to Wavelets*, Academic Press, New York (1992).
14. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Soc. Conf. Computer Vision Patt. Recog.* **1**, 511–518 (2001).
15. R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. Intl. Conf. Image Process.* **1**, 900–903 (2002).
16. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York (1966).
17. "USC-SIPI image database–textures," see http://sipi.usc.edu/database/database.cgi?volume=textures (2009).
18. S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on gabor filters," *IEEE Trans. Image Process.* **11**(10), 1160–1167 (2002).
19. F. Xu and Y. J. Zhang, "Evaluation and comparison of texture descriptors proposed in mpeg-7," *J. Visual Commun. Image Represent.* **17**(4), 701–716 (2006).
20. J. K. Kämäräinen, V. Kyrki, and H. Kälviäinen "Invariance properties of gabor filter-based features-overview and applications," *IEEE Trans. Image Process.* **15**(5), 1088–1099 (2006).
21. P. Kruizinga, N. Petkov, and S. E. Grigorescu, "Comparison of texture features based on gabor filters," in *Proc. 10th Intl. Conf. Image Anal. Process.*, p. 142, Washington, DC (1999).
22. P. Kruizinga and N. Petkov, "Nonlinear operator for oriented texture," *IEEE Trans. Image Process.* **8**(10), 1395–1407 (1999).
23. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York (1990).
24. R. J. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*, Wiley, New York (1992).
25. "Outex texture database," see http://www.outex.oulu.fi/ (2009).

26. C. M. Pun and M. C. Lee, "Log-polar wavelet energy signatures for rotation and scale invariant texture classification," *IEEE Trans. Patt. Anal. Mach. Intel.* **25**(5), 590–603 (2003).
27. "FFTW package for DFT," see http://www.fftw.org.

Kam-Keung Fung received his BASc in engineering physics and MASc in electrical engineering from the University of British Columbia, Canada, in 1992 and 1995, respectively. He has recently completed his DEng at the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. Professionally, he has conducted applied research for medical devices, and developed and managed consumer electronic products. His current research interests include image processing and pattern recognition.

Kin-Man Lam received his Associateship in electronic engineering with distinction from The Hong Kong Polytechnic University in 1986. He won the S.L. Poa Scholarship for overseas studies and was awarded an MSc degree in communication engineering from the Department of Electrical Engineering, Imperial College of Science, Technology, and Medicine, England, in 1987. In 1993, he undertook a PhD program in the Department of Electrical Engineering at the University of Sydney, Australia, and won an Australia Postgraduate Award for his studies. He completed his PhD studies in 1996, and was awarded the IBM Australia Research Student Project Prize. From 1990 to 1993, he was a lecturer at the Department of Electronic Engineering of The Hong Kong Polytechnic University. He joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University again as an assistant professor in 1996, and became an associate professor in 1999, and has been a professor since 2010. His current research interests include human face recognition, image and video processing, and computer vision.