Texture classification and discrimination for region-based image retrieval

Mohsen Zand *, Shyamala Doraisamy, Alfian Abdul Halin, Mas Rina Mustaffa

Department of Multimedia, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

A R T I C L E   I N F O

Article history:
Received 1 April 2014
Accepted 4 October 2014
Available online 20 October 2014

Keywords:
Region-based image retrieval
Texture feature extraction
Texture classification
Gabor wavelet
Curvelet filters
Polynomials
ImageCLEF
Outex

A B S T R A C T

In RBIR, texture features are crucial in determining the class a region belongs to since they can overcome the limitations of color and shape features. Two robust approaches to model texture features are Gabor and curvelet features. Although both features are close to human visual perception, sufficient information needs to be extracted from their sub-bands for effective texture classification. Moreover, shape irregularity can be a problem since Gabor and curvelet transforms can only be applied on the regular shapes. In this paper, we propose an approach that uses both the Gabor wavelet and the curvelet transforms on the transferred regular shapes of the image regions. We also apply a fitting method to encode the sub-bands' information in the polynomial coefficients to create a texture feature vector with the maximum power of discrimination. Experiments on texture classification task with ImageCLEF and Outex databases demonstrate the effectiveness of the proposed approach.

© 2014 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

The rapid growth of image data on the internet has spurred the demand for methods and tools for efficient search and retrieval. Although many researches have been done in the field of image search and retrieval, there are still many challenging problems to be solved. As the semantic gap is considered to be the main issue, recent works have focused on semantic-based image retrieval. Most of the proposed approaches learn image semantics by extracting low-level features from entire image. However, such approaches fail to take into consideration the semantic concepts that occur in the images. In this paper, we focus on the high-level semantic identification at the region level. This is because analyzing the visual features included in the images gives more intuition about images. By learning these features at the region level, high-level semantics of images can be built. The approaches in which region information is employed to extract semantic concepts of images are known as region-based image retrieval or RBIR [1–3].

One issue in semantic understanding of image regions in RBIR is the extraction of effective and discriminatory features. Many researches have been done in global image features extraction and representation, but not much attention has been paid to region-based features [2–6]. Ideally, the extracted features must match the human perceptions of images. Most image retrieval systems apply three well-known color, shape and texture features. Color is the most common feature since it is invariant to distortion and scale. Although color feature is well-defined and widely used in image retrieval systems, it is unable to distinguish between different objects with the same color. Shape feature is not as important as other features in RBIR as regions’ shapes are more vulnerable than regions’ color and texture features [2,7].

Texture is an important determinant of region class in RBIR due to its capability to distinguish regions with similar colors and shapes. Although texture feature is very useful in RBIR, they are difficult to model. Ideally, a texture feature of an object should be consistent with human perceptual intuitions of the object, like directional/chaotic and smooth/rough [8].

Basically, texture analysis has four main categories: (1) texture feature, (2) texture discrimination, (3) texture classification, and (4) shape from texture. In this paper, we consider only texture discrimination and classification. Many different methods have been proposed for texture feature extraction. In general, these methods are categorized into spatial and spectral approaches [9,10].

The spatial approaches are further classified into structural, statistical and model-based approaches. In structural approaches such as Voronoi tessellation [11], texture feature is described using a set of texture primitives and their placement rules. Statistical texture...
features are usually based on low level statistics of grey level co-
occurrence matrices (GLCM) [12,13]. Although these features are
compact and robust, they are insufficient to describe a large variety
of textures. Model-based approaches such as Markov random fields
(MRF) [14] and fractal dimensions (FD) [15] apply stochastic (ran-
dom) or generative models to describe texture features. As these
models fall into an optimization problem, they usually need com-
plex computations [16].

In spectral methods such as discrete cosine transform [18], Fou-
rier transform [19], wavelet filters [20], Gabor [21] and curvelet
features [22], texture images are transformed into the frequency
domain using a set of spatial filters. Then, the statistics of the spec-
tral information at different scales and orientations form the tex-
ture descriptor. Due to the large neighborhood support of the
filters, spectral methods can generate sufficient number of features
to classify variety of texture images.

However, the varying rotations of real-world textures suggest
the need for rotation–invariant methods. Among the many spatial
methods that can be considered, LBP [17] is the most widely used.
It combines structural and statistical approaches by computing the
occurrence histogram for rotation–invariant texture classification.
In LBP, the values of neighboring pixels are turned into binary val-
ues using the central pixel as the threshold. This local binary gray-
scale information is encoded to characterize a structural pattern.
Although rotation invariance is achieved by only selecting
rotation–invariant uniform local binary patterns, it is not scale-
invariant. The LBP-based approaches often also fail in detecting
large-scale textural structures. Many Gabor- and wavelet-based
algorithms were also proposed for rotation–invariant texture
classification [23–25]. Han and Ma [25] proposed to create texture
features from a rotation–invariant and a scale-invariant Gabor rep-
resentation by summations of the conventional Gabor filters.
Recently, [2,26] presented a circular shifting of the curvelet texture
features to generate rotation–invariant texture representations.

However, both the Gabor wavelet and the curvelet filters cap-
ture a large volume of unnecessary information which reduces
their distinguishing power in texture classification. To overcome
this issue, sub-band coefficients are produced in multiple orienta-
tions and scales and analyzed in the pre-processing step. In some
earlier works, generalized Gaussian density was used to model
wavelet coefficients [20,27,28]. In most of the texture extraction
methods, texture feature vectors consist of statistical information,
which are calculated from all sub-bands generated by applying
either Gabor or curvelet transforms on a given image [23,29,30].
Zhang et al. [2] used mean and standard deviation to create a tex-
ture feature vector from curvelet sub-bands for each image region.
Although both Gabor and curvelet transforms represent image tex-
ture features sufficiently by sub-band coefficients, using mean and
standard deviation and other statistical information can lead to
misclassifications as the discrimination power reduces. For
instance, each image pair in Fig. 1 shows two different textures
being classified into the same group as they have similar or slightly
different means and standard deviations.

Mohamadzadeh and Farsi [7] used down-sampling to create a
reduced size feature vector. However, random down-sampling
increases the risk of losing the key information in the respective
sub-band.

To overcome these problems, we propose the application of
polynomial coefficients that are unique in representing data points.
It is a combination of sub-band coefficients computed by the Gabor

![Fig. 1. Pairs of different textures with similar statistical values obtained from curvelet sub-bands, which lead to the misclassification.](image1)

![Fig. 2. The block diagram of the proposed method.](image2)
wavelet and curvelet transforms. Using polynomial coefficients can therefore classify similar textures in the same group which leads to effective discrimination of different textures when both Gabor and curvelet features are combined. The block diagram of the proposed texture classification method is shown in Fig. 2.

Experiments show that the proposed method performs well in comparison to the several other methods, namely that of Zhang et al.’s [2], LBP [17] and systems that only use the Gabor wavelet or curvelet features.

The rest of the paper is organized as follows. In Section 2, we describe the basic concepts required for region-based texture classification. In Section 3, we propose a method to combine both Gabor wavelet and curvelet transforms and create texture feature vector by polynomial coefficients. In Section 4, the evaluations and comparisons are presented. Finally, Section 5 summarizes and concludes the paper.

2. Basic concepts

In this section, the definitions of Gabor wavelet, curvelet transform and polynomial function required for region-based texture classification are described.

2.1. Gabor wavelet filter

The Gabor transform is a type of wavelet filter. According to [14], a two dimensional Gabor function $g(x, y)$ can be written as:

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

where $W$ is the modulation frequency. The 2-D Fourier transform which is the frequency response of the Gabor function is then defined as:

$$G(u, v) = \exp \left[ -\frac{1}{2} \left( \frac{u^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right) \right],$$

where $\sigma_u = (2\pi \sigma_x)^{-1}$ and $\sigma_v = (2\pi \sigma_y)^{-1}$. Let $g(x, y)$ be the mother Gabor wavelet, then a set of self-similar filter functions can be generated by appropriate dilations and rotations of $g(x, y)$ through the following generating function:

$$g_{mn}(x, y) = a^{-m}g(x', y'),$$

where $x' = a^{-m}(x \cos \theta + y \sin \theta), y' = a^{-m}(-x \sin \theta + y \cos \theta), a > 1, \theta = n\pi/N$, and $n = 0, 1, 2, \ldots, N - 1$ and $m = 0, 1, 2, \ldots, M - 1$ are the orientation and the scale of the Gabor wavelet, respectively. Accordingly, for a given image region $I(x, y)$ with the size $X \times Y$, its discrete Gabor wavelet transform is computed as:

$$G_{mn}(x, y) = \sum_{i} \sum_{j} I(i, j) g_{mn}^*(x - i, y - j),$$

where $^*$ indicates the complex conjugate of $g_{mn}$. The Gabor wavelet filter is then applied on the image region in different orientations and scales to obtain an array of magnitudes:

![Fig. 3. Frequency spectrum coverage by (a) Gabor (b) curvelet.](image)

![Fig. 4. (a) A ‘leaf’ region image. (b) One curvelet sub-band computed from image (a). (c) Fitted polynomial of (b). (d) Function and coefficients of the polynomial in (c).](image)
These magnitudes are the sub-bands which represent the energy contents at several scales and orientations for a given region. It has been proven that they are descriptive image texture features similar to how human vision is invariant to scale and orientation.

2.2. Curvelet filter

A curvelet transform is an extension of the ridgelet transform. The ridgelet differs from other wavelet transforms in which it detects lines instead of points. According to [31], a continuous ridgelet transform at scale $a$, translation $b$, and orientation $\theta$ is given by:

$$\text{CRT}_f(a, b, \theta) = \int \int w_{a,b,h}(x,y) I(x,y) \, dx \, dy,$$

where $I(x,y)$ is an image region, and $\psi$ is the ridgelet function which is defined as:

$$\psi_{a,b,h}(x,y) = a^{-1/2} \psi \left( \frac{x \cos \theta + y \sin \theta - b}{a} \right).$$

Curvelet feature extraction function for digital image region $I[x,y]$ can then be computed as:

$$E(m,n) = \sum_x \sum_y |G_{mn}(x,y)|.$$  (5)
CT_{a,b,\theta} = \sum_{0<i<m} \sum_{0<j<n} I(x,y)\psi_{a,b,i,j}(x,y).

(8)

As explained in [32], this formula can be extended in the frequency domain and defined as:

CT_{a,b,\theta} = \text{IFFT}(\text{FFT}(I(x,y)) \times \text{FFT}(\psi_{a,b,i,j}(x,y)))

(9)

The curvelet sub-bands for every region are created by adjusting the ridgelets to different scales and orientations. Then, the energy of each curvelet sub-band can be computed as:

\[ E(a, \theta) = \sum_{x,y} |S_{a,b}(x,y)|^2 \]

(10)

where \(S_{a,b}\) is the sub-band at scale \(a\) and orientation \(\theta\).

Though Gabor filters are similar in this way, they do not cover entire frequency spectrums due to their oval shape. Fig. 3(a) represents the holes between ovals in the frequency plane of Gabor wavelet. The complete coverage of the frequency spectrum by curvelet can be observed in Fig. 3(b). In this figure, \(s_i\) stands for scale \(i\), and the numbers 1, 2, . . ., etc. show the orientations or sub-bands.

Although curvelet can cover spectral domain completely due to the wedge shape of its frequency response, the Gabor wavelets are very effective in representing objects with isolated point singularities. Thus, Gabor wavelets and curvelets are mutually complementary. Recently, the combination of wavelet and curvelet transforms has been reported in some applications for image denoising and feature extraction [33–36], which their findings suggest the potential of the combination. Thus, we apply both methods to create texture feature vectors for the best classification accuracy.

2.3. Fitting by polynomials

Polynomial functions have flexible shapes, and changes of location and scale in the raw data result in the same polynomial model. That is, the underlying metrics do not influence the polynomial functions. They are frequently applied as an empirical method for curve and surface fitting, so they can encode information about some other objects. A polynomial function is a data set of \(n\) paired \((x,y)\) members such as \((x_1,y_1),(x_2,y_2),(x_3,y_3),\ldots,(x_n,y_n)\), which are evaluated using a least-squares technique to generate a predictive polynomial equation \(y\) as illustrated in the following:

\[ y = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots + a_px^p, \quad p < n. \]

(11)

where \(p\) stands for the degree of the polynomial, which is a non-negative integer. In this equation, \(a_0, a_1, \ldots, a_p\) are the polynomial coefficients, which are constants. These unknown coefficients are calculated by minimizing the sum of the deviations' squares of the data from the model, which is called least-squares fit. This mathematical procedure is described in detail in [37].

Multivariate polynomial functions can also be defined. Polynomials in two variables are algebraic expressions taking the form of \(ax^my^n\) instead of \(ax^n\). The degree of these polynomials is the largest sum of the exponents in all terms.

In this paper, we use a polynomial function with two variables to encode each Gabor or curvelet sub-band in the spectral domain of an image region texture. The following equation has been created for a curvelet transform sub-band. The constants \(p_{00}, p_{10}, p_{01}, p_{20}, p_{11}, p_{02}, p_{30}, p_{21}, p_{12}\), and \(p_{03}\) are coefficients which properly code the information in the respective sub-band:

\[ f(x,y) = p_{00} + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy + p_{02}y^2 + p_{30}x^3 + p_{21}x^2y + p_{12}xy^2 + p_{03}y^3. \]

(12)

Fig. 4 illustrates a texture and its corresponding curvelet sub-band which has been fitted by a polynomial function.

3. Texture classification

Prior to texture classification task, images in the training set are segmented into regions using the JSEG in [38]. Since image regions are irregular, they must be transferred to a regular shape in order to apply Gabor wavelet and curvelet transforms.

An irregular shaped region is transformed into an appropriate regular shaped region for texture extraction by finding either the largest internal square or the bounding box. Due to the size variation in the regions, the largest internal square of some regions may not include enough texture information. Although a bounding box is large enough, but it always includes non-region pixels. The accuracy of the extracted texture features from a bounding box highly depends on the values of these pixels. The common approach to fill these regions is via ‘zero-padding’, where non-region pixels are filled by zeros. Nevertheless, differences between zero padded regions and the original region result in false and inaccurate texture information. Recently, Zhang et al. [2] proposed a mirror padding method to fill non-region areas in the bounding box with the mirrored textures inside the region. Although this method outperforms zero padding, it adds false information of the region edges to the real texture of the region. In contrast, an internal square consists of only valid region-pixels. Thus the features extracted from an internal square are more accurate than the features extracted from a bounding box. Thus we propose an improved version of the mirror padding which extract both real and mirrored textures from the largest internal square instead of the bounding box. It also avoids mirror padding when the size of the largest internal square is large enough to carry texture information of the region. Fig. 5 shows the wrong reflected region edges using mirror padding method, which have been removed in the improved mirror padding method.

Once all the training image regions are transformed into the regular shapes using the improved mirror padding method, they are categorized into predefined concept classes according to their textures. For a particular concept with a distinct texture, multiple regular shaped regions are gathered to create an image region.

Fig. 7. A texture image in the three rotation angles (a) 0°, (b) 15° and (c) 30°.
Table 1

<table>
<thead>
<tr>
<th>0° Fig. 7(a)</th>
<th>15° Fig. 7(b)</th>
<th>30° Fig. 7(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>Rearranged</td>
<td>Original</td>
</tr>
<tr>
<td>556.24</td>
<td>558.76</td>
<td>498.72</td>
</tr>
<tr>
<td>523.00</td>
<td>524.35</td>
<td>535.10</td>
</tr>
<tr>
<td>466.74</td>
<td>462.25</td>
<td>549.70</td>
</tr>
<tr>
<td>455.04</td>
<td>457.69</td>
<td>519.81</td>
</tr>
<tr>
<td>478.99</td>
<td>469.10</td>
<td>467.28</td>
</tr>
<tr>
<td>558.76</td>
<td>483.39</td>
<td>452.87</td>
</tr>
<tr>
<td>524.35</td>
<td>548.31</td>
<td>473.49</td>
</tr>
<tr>
<td>462.25</td>
<td>555.09</td>
<td>550.07</td>
</tr>
<tr>
<td>557.69</td>
<td>498.77</td>
<td>524.83</td>
</tr>
<tr>
<td>469.10</td>
<td>496.79</td>
<td>467.26</td>
</tr>
<tr>
<td>483.39</td>
<td>543.74</td>
<td>461.23</td>
</tr>
<tr>
<td>548.31</td>
<td>556.24</td>
<td>469.83</td>
</tr>
<tr>
<td>555.09</td>
<td>523.00</td>
<td>485.80</td>
</tr>
<tr>
<td>498.77</td>
<td>466.74</td>
<td>545.05</td>
</tr>
<tr>
<td>469.79</td>
<td>455.04</td>
<td>548.89</td>
</tr>
<tr>
<td>543.74</td>
<td>478.99</td>
<td>501.34</td>
</tr>
</tbody>
</table>

The sub-band energies of the rotated textures in Fig. 7 in one scale, and their rearranged energies.

Table 2

<table>
<thead>
<tr>
<th>0° Fig. 7(a)</th>
<th>15° Fig. 7(b)</th>
<th>30° Fig. 7(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Rearranged Original Rearranged Original Rearranged</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.16</td>
<td>1.17</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.86</td>
<td>-0.64</td>
<td>-0.63</td>
</tr>
<tr>
<td>0.62</td>
<td>9.56</td>
<td>10.13</td>
</tr>
<tr>
<td>0.66</td>
<td>0.78</td>
<td>0.81</td>
</tr>
<tr>
<td>-1.00</td>
<td>-1.67</td>
<td>-1.74</td>
</tr>
<tr>
<td>0.88</td>
<td>0.89</td>
<td>1.05</td>
</tr>
<tr>
<td>4.82</td>
<td>5.10</td>
<td>5.21</td>
</tr>
<tr>
<td>1.06</td>
<td>1.08</td>
<td>1.18</td>
</tr>
<tr>
<td>1.98</td>
<td>1.96</td>
<td>1.93</td>
</tr>
<tr>
<td>1.04</td>
<td>0.91</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Coefficients of the fitted polynomials of the textures in Fig. 7.

Our proposed algorithm employs 40 Gabor filters in five scales and eight orientations as shown in Fig. 6(b) where we illustrate the texture detection steps along with the image region after applying Gabor filters. We also apply curvelet decomposition on the image texture detection steps along with the image region after applying Gabor and curvelet transforms of various textures in the same class can be coded into similar codewords with the coefficients of their fitted polynomials. Fig. 8 illustrates two different textures for the class ‘grass’ with very similar coefficients of their fitted curvelet sub-bands. The higher discrimination power is another benefit of using polynomials. It is also observed that the sub-bands obtained from the Gabor and curvelet transforms of various textures in the same class can be used in texture classification and discrimination. First, same textures have the same coefficients after fitting the polynomials. It is also observed that the sub-bands obtained from the Gabor and curvelet transforms of various textures in the same class can be coded into similar codewords with the coefficients of their fitted polynomials. This work, given that the energy of the dominant orientation usually spreads between two neighboring sub-bands, different sub-bands at each scale are rearranged based on their energies. This makes a unique sub-band set for each texture, even though it is rotated, and therefore, encoding a texture at different rotations generates similar coefficients. More specifically, although both the Gabor and the curvelet coefficients reflect the effect of the rotation at different scales, the rotation of the texture cannot be captured by only considering these coefficients. However, there is a dominant orientation at each scale of decomposition in the Fourier spectrum which is detectable by finding the highest energy sub-band. Consequently, rearranging the other sub-bands makes a relatively unique set of coefficients for each texture even though it is rotated. Using Eqs. (5) and (10), the highest energy sub-bands can be detected.

Fig. 7 shows a texture image at different rotation angles of 0°, 15°, and 30°, which their sub-band energies are listed in Table 1. It is observable that these energies are concentrated at the dominant values and they are circularly shifted when the texture is rotated. Rearranging the respective sub-bands and fitting them by polynomials (using Eq. (12)) result in a unique feature vector consisting of the polynomial coefficients as shown in Table 2. To encode the rearranged coefficients using polynomial fitting, first we obtain the degree of the bivariate polynomial that best fits the data. From our preliminary experiments, this varies between 3° and 6° for different sub-bands. We finally choose the value 6 so that the feature vectors with the same size are created for all regions and sub-bands.

Using polynomials to encode texture information brings several benefits in texture classification and discrimination. First, same textures have the same coefficients after fitting the polynomials. It is also observed that the sub-bands obtained from the Gabor and curvelet transforms of various textures in the same class can be coded into similar codewords with the coefficients of their fitted polynomials. Fig. 8 illustrates two different textures for the class ‘grass’ with very similar coefficients of their fitted curvelet sub-band polynomials. The higher discrimination power is another benefit of using polynomials. It is described by the Fisher's linear discriminant or FLD which is a widely used discrimination criterion that measures the “between-class scatter” normalized by the “within-class scatter” [39]. It tries to maximize the scatter ratio between classes to the scatter within classes. Let us assume we have sets Xi, i = 1, 2, 3…, c which represent c classes, each containing ni elements that |Di| = ni. The scatter within classes is given by:

\[ S_W = \sum S_{0i} \]
where $S_{D_i}$ is the sum of all classes scatters and is defined as:

$$S_{D_i} = \sum_{x \in D_i} (x - m_{D_i})(x - m_{D_i})^T,$$

which $m_{D_i}$ is the mean vector for class $D_i$. The total mean that is in fact the centers of gravity of all classes means is defined as:

$$m = \frac{1}{N} \sum_{i=1}^{c} n_i m_{D_i}.$$  \hspace{1cm} (15)

The scatter between classes is also computed by:

$$S_B = \sum_{i=1}^{c} n_i (m_{D_i} - m)(m_{D_i} - m)^T.$$ \hspace{1cm} (16)

And finally, we define the discrimination as following:

$$D_{is} = \frac{S_B}{S_{W}}.$$ \hspace{1cm} (17)

As a result, the higher values of this metric show lower scatter within classes and higher scatter between classes, which result in a better discrimination. This leads to compact and well-separated clusters.

In the next step, all texture features in each training texture class are used to create the representative feature vector for that class, by averaging all vectors in the texture category. We then build up a texture dictionary from these feature vectors where every entry includes a codeword to represent a particular texture. A texture dictionary is a set of codewords which are representative texture features. The number of entries in the texture dictionary demonstrates the number of texture classes. In other words, every region texture class is coded into a codeword in the texture dictionary. These codewords are generated using the vector quantization method which discretizes texture feature vectors and computes the centroid of each cluster by averaging the cluster of textures [40].

Once the texture dictionary is created, a mapping between a semantic concept and codewords from the dictionary needs to be established. Therefore we applied a $k$-nearest neighbor ($k-NN$) classifier based on the proposed classification method in [41] to perform classification step. This classifier projects the extracted region texture to the appropriate texture class according to the texture descriptors of the regions. To make predictions with $k-NN$, we measure the distance between the texture vector and the codewords using the Euclidean distance. We find the index of the closest representative feature vector from the texture dictionary by:

$$\text{ind} = \arg\min_{i} \text{dist}(v, v_i),$$ \hspace{1cm} (18)

where $i$ is an integer between 1 and the dictionary size, and $\text{dist}(v, v_i)$ is the Euclidean distance between the texture vector and the codewords from the texture dictionary.

### 4. Experimental results

Two experiments with differing settings were conducted to demonstrate the performance of the proposed method in terms of scale and rotation invariance. The first experiment, discussed in Section 4.1, investigates the effectiveness of the proposed method in a challenging texture classification task of rotation invariance. It also considers the empirical results on different polynomial degrees to find the optimal degree. In this experiment, our results are compared with the widely used rotation–invariant texture classification methods by Ojala et al. [17], and Zhang et al. [2] using the standard and popular Outex [42] database.

<table>
<thead>
<tr>
<th>Polynomial degree</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>54</td>
<td>73</td>
<td>91</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3 Comparison of classification scores for different polynomial degrees.
The second experiment, discussed in Section 4.2, aims at considering the proposed method in classifying the real textures of real-world images from the ImageCLEF 2012 dataset [43] for RBIR, enabling evaluation at varying scales. In this experiment, the performance of the proposed method is evaluated and compared with the proposed method by Zhang et al. [2] and systems that only use the Gabor wavelet or the curvelet features.

### 4.1. Experiments on classifying textures in different rotations

We first investigated the optimal degree for our polynomial function, and consequently used the results through the experiments. To verify that the proposed texture descriptor is not only scale-invariant but also achieves better rotation invariance in comparison to the other methods, a series of experiments based on the Outex database [42] was carried out. Outex includes a large collection of textures and test suites for different types of texture analysis challenges. For fair comparison, we used the same texture images and followed the experimental setup by Ojala et al. [17]. Consequently, the Outex\_TC\_00010 test suite was selected as a standard benchmark to evaluate the rotation invariance of the texture classification method. It contains 24 texture classes, as shown in Fig. 9, with 180 images of size 128 \times 128 in 9 rotation degrees in each class (20 samples for each rotation degree). Each 128 \times 128 texture sample was normalized to an average intensity of 128 and a standard deviation of 20. For the purpose of classification, the k – NN classifier was used, with the Euclidean distance as the similarity measurement between two descriptors.

To objectively evaluate the percentage of correctly classified samples in the same way for all algorithms applying on Outex, a performance metric i.e. the classification score \( S \) is defined by:

\[
S = \frac{\sum_{n=1}^{N} \delta(L_{n}, N_{\text{th}}) C_{n}^k}{\sum_{n=1}^{N} C_{n}^k} \times 100, \tag{19}
\]

where \( N \) is the number of test samples, \( \delta \) is Kronecker delta function, \( C_{n}^k \) is the predefined cost for the \( n \)-th sample in the test set, and \( L_{n}\) and \( L_{N_{\text{th}}} \) are the original texture class and the assigned texture class of the \( n \)-th sample in the test set, respectively.

In order to obtain an optimum degree for the polynomial function, the classifier was trained with randomly selected samples from all rotation degrees; five samples of each rotation degree in each class (5 \times 9 samples). The remaining 3240 samples were used for testing the classification accuracy of different polynomial degrees varying between 3 and 6. As shown in Table 3, there is a trend of increasing classification performance as the polynomial degree increases. Surprisingly, all test samples were classified in their appropriate texture classes when the polynomial degree of 6 was selected. However, the degree 6 was chosen regarding the tradeoff between the accuracy and the feature size of the higher polynomial degrees. This result not only suggests the optimal polynomial degree for our method but also verifies the strength of the polynomial coefficients for texture discrimination. This setting was then used in the other experiments.

After obtaining the optimal polynomial degree, we trained the classifier in different ways for evaluating the rotation invariance of the algorithm. In this setting, only the samples at rotation angle 0° were applied to train the classifier, and samples with the other 8 angles (5°, 10°, 15°, 30°, 45°, 60°, 75°, 90°) in each texture class were used as unseen test sets. This resulted in a total of 480 samples for training and 3840 samples for validation. Table 4 compares the classification scores of the proposed method with those of LBP [17] and Statistical curvelet [2] in different rotation angles.

While the polynomial fitting method outperformed the two other methods in multi-rotation classification case, the statistical curvelet [2] obtained the lower mean score of 79.22%. This is because much useful texture information is kept in our encoding process, and rotation–invariant features are efficiently generated by normalizing the circular shifted feature vector of each sub-band.

To observe the effect of the rotation in different texture classes, the misclassification results for both methods showing the highest scores of polynomial fitting and LBP are listed in Table 5. In our case, very similar textures apparently obtained the highest number of misclassified samples in different rotation angles. Hence, the two classes ‘convas038’ and ‘convas039’, as well as ‘tile005’ and ‘tile006’ were encoded to the similar texture features due to their theme similarity, and this also caused them to be wrongly classified as each other in some rotation angles. For instance, when ‘convas039’ rotates by 5° or 10° degrees, it is clearly much similar to ‘convas038’. It is also observable that the LBP faces more difficulty in classifying ‘convas033’, which shows inefficiency of LBP in detecting the joint analysis of the large-scale textural structures.

### 4.2. Experiments on the real image regions


The images used for the training and testing phases were from the ImageCLEF 2012 dataset [43], which itself was made from the MIRFLICKR-1M dataset [44]. We used 1000 image regions to create the texture dictionary and the classifiers. The remaining 1000 regions were used to test the classification accuracy. Our method was compared with the texture classifiers using only Gabor wavelet, only curvelet transform, and the statistical curvelet method proposed by Zhang et al. [2].

In the training phase, output regions of the JSEG tool were transferred to the regular shapes with the size of 64 \times 64 either using improved mirror padding method or only the pre-processing step if their largest internal squares were greater than 40 \times 40. We chose this size because it is large enough to represent region texture. The texture features from the training regions were then processed to extract the Gabor and curvelet representations. The Gabor texture representation consisted of 40 different sub-bands. Each sub-band was fitted into a polynomial, and its coefficients created the Gabor texture representation. This together with the coefficients of the fitted polynomials of the curvelet representation sub-bands created the texture feature vector. The curvelet sub-bands describe a region texture in 4 levels of curvelet decomposition, which results in 50(=1 \times 16 + 32 + 1) sub-bands. All 1000 training texture feature vectors obtained from the Gabor and curvelet transforms were used to create codewords in the texture.

### Table 4

Comparison of classification scores in different rotation angles for different methods.

<table>
<thead>
<tr>
<th>Rotation angle</th>
<th>0°</th>
<th>5°</th>
<th>10°</th>
<th>15°</th>
<th>30°</th>
<th>45°</th>
<th>60°</th>
<th>75°</th>
<th>90°</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical curvelet [2]</td>
<td>100</td>
<td>84.05</td>
<td>78.11</td>
<td>76.46</td>
<td>75.00</td>
<td>75.92</td>
<td>75.63</td>
<td>74.73</td>
<td>73.12</td>
<td>79.22</td>
</tr>
<tr>
<td>LBP [17]</td>
<td>100</td>
<td>99.07</td>
<td>97.95</td>
<td>96.88</td>
<td>97.75</td>
<td>96.57</td>
<td>97.93</td>
<td>95.58</td>
<td>94.53</td>
<td>97.36</td>
</tr>
<tr>
<td>Fusion of Gabor and curvelet with fitted polynomial</td>
<td>100</td>
<td>96.5</td>
<td>97.66</td>
<td>98.04</td>
<td>99.18</td>
<td>99.18</td>
<td>98.46</td>
<td>97.64</td>
<td>98.75</td>
<td>98.38</td>
</tr>
</tbody>
</table>
dictionary. After the codewords were created, a $k - \text{NN}$ classifier was used to classify the remaining test samples.

The classification accuracy for only the Gabor feature, only the curvelet feature, fusion of Gabor and curvelet features, and statistical curvelet methods are given in Table 6. Each entry specifies the average classification rate for every region texture class which is the percentage of the regions correctly classified between all regions in that class.

From Table 6, it can be seen that the classification accuracy is at its best when both the Gabor and the curvelet features are considered and fitted by polynomial coefficients. The two other methods that use curvelets perform better than the Gabor wavelet. The lowest percentage of 59% is obtained by the Gabor wavelet method. Therefore, it can be concluded that curvelet features are more representative of the image textures than the Gabor wavelets as reported in [45]. This is because curvelet features are more successful in representing complex textures than Gabor wavelets as they capture additional directional information in the spectral domain. They also efficiently capture the edge information at high frequency levels which is a noteworthy texture discriminating property. However, extracting statistical information such as mean and standard deviation from curvelet sub-bands as suggested by Zhang et al. [2] decreases the texture classification rate. It should also be noted that although the Gabor wavelet suffers from the

<table>
<thead>
<tr>
<th>Texture classes</th>
<th>Classification rate (%)</th>
<th>Gabor wavelet with fitted polynomial</th>
<th>Curvelet with fitted polynomial</th>
<th>Fusion of Gabor and curvelet with fitted polynomial</th>
<th>Statistical curvelet [2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beach</td>
<td>63</td>
<td>65</td>
<td>67</td>
<td>67</td>
<td>65</td>
</tr>
<tr>
<td>Branch</td>
<td>43</td>
<td>67</td>
<td>76</td>
<td>76</td>
<td>72</td>
</tr>
<tr>
<td>Bush</td>
<td>50</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>63</td>
</tr>
<tr>
<td>City</td>
<td>42</td>
<td>60</td>
<td>80</td>
<td>80</td>
<td>78</td>
</tr>
<tr>
<td>Cloud</td>
<td>49</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>75</td>
</tr>
<tr>
<td>Giraffe</td>
<td>51</td>
<td>51</td>
<td>67</td>
<td>67</td>
<td>51</td>
</tr>
<tr>
<td>Glacier</td>
<td>86</td>
<td>86</td>
<td>86</td>
<td>86</td>
<td>57</td>
</tr>
<tr>
<td>Grass</td>
<td>42</td>
<td>60</td>
<td>80</td>
<td>80</td>
<td>48</td>
</tr>
<tr>
<td>Highway</td>
<td>55</td>
<td>82</td>
<td>91</td>
<td>91</td>
<td>45</td>
</tr>
<tr>
<td>Lake</td>
<td>45</td>
<td>36</td>
<td>82</td>
<td>82</td>
<td>64</td>
</tr>
<tr>
<td>Light</td>
<td>89</td>
<td>91</td>
<td>95</td>
<td>95</td>
<td>93</td>
</tr>
<tr>
<td>Orange</td>
<td>80</td>
<td>60</td>
<td>80</td>
<td>80</td>
<td>53</td>
</tr>
<tr>
<td>Sand-beach</td>
<td>83</td>
<td>67</td>
<td>83</td>
<td>83</td>
<td>50</td>
</tr>
<tr>
<td>Sky-blue</td>
<td>57</td>
<td>75</td>
<td>92</td>
<td>92</td>
<td>48</td>
</tr>
<tr>
<td>Sky-night</td>
<td>57</td>
<td>79</td>
<td>93</td>
<td>93</td>
<td>71</td>
</tr>
<tr>
<td>Sunset-dusk</td>
<td>57</td>
<td>86</td>
<td>86</td>
<td>86</td>
<td>54</td>
</tr>
<tr>
<td>Snow</td>
<td>51</td>
<td>55</td>
<td>80</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>Sun</td>
<td>61</td>
<td>85</td>
<td>89</td>
<td>89</td>
<td>78</td>
</tr>
<tr>
<td>Wall</td>
<td>76</td>
<td>67</td>
<td>76</td>
<td>76</td>
<td>75</td>
</tr>
<tr>
<td>Waves</td>
<td>51</td>
<td>56</td>
<td>75</td>
<td>75</td>
<td>72</td>
</tr>
<tr>
<td>Mean of all categories</td>
<td>59</td>
<td>70</td>
<td>82</td>
<td>82</td>
<td>64</td>
</tr>
</tbody>
</table>
problem of directional information loss, combining it with the curvelet features increases the classification rate dramatically. It is observable that the classification rate for the statistical curvelet method is 64%, whereas the highest percentage of 82% is obtained using our proposed method where both the Gabor and the curvelet features are used and fitted by polynomials. Fig. 10 shows classification accuracy of different texture categories. Fig. 11 illustrates the image regions in different texture categories which are correctly classified using our proposed classification method.

Our proposed method also indicated the best discrimination power among studied methods. Table 7 shows the discrimination powers of these methods, which have been obtained using normalization of Eq. (17). It can be observed that mean and standard deviation are not as strong as polynomial coefficients in discriminating different textures. There is a slight difference between Gabor, Curvelet and their fusion in discriminating textures when all use polynomial coefficients.

To evaluate the retrieval performance of the proposed method, the precision-recall graph is shown for the 2000 image region textures from the ImageCLEF database. Precision is the fraction of retrieved images that are relevant, and recall is the fraction of relevant images that are retrieved. Each region with its known
texture class was used as a query, and its precisions were calculated at 10 levels of recall percentages. The average precisions for all regions at each recall level, and for all tested methods are plotted in Fig. 12. This figure shows that fusion of the Gabor and the curvelet with fitted polynomials outperforms the other methods. It also verifies that simple sampling of the curvelet sub-bands degrades the performance as the statistical curvelet obtained the lowest average precisions for values of recall greater than 30%.

5. Conclusion

The paper investigates the problem of texture classification in RBIR. It considers the combination of the two most robust texture representation methods, Gabor wavelet and curvelet filters. The proposed method takes the advantages of both and uses polynomial coefficients to encode their sub-bands information in the texture feature vectors. Using polynomial is based on the idea that produced sub-bands of these filters are surfaces that can be fitted by polynomials uniquely. This reduces the texture feature space, increases the texture classification rate and produces more discrimination power. However, the performance of the RBIR systems based on this method can be further improved by employing other features like color and shape.

References


Table 7

Comparison of discrimination power between methods.

<table>
<thead>
<tr>
<th>Texture classification methods</th>
<th>Gabor wavelet with fitted polynomial</th>
<th>Curvelet with fitted polynomial</th>
<th>Fusion of Gabor and curvelet with fitted polynomial</th>
<th>Statistical curvelet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrimination power</td>
<td>79.3</td>
<td>80.6</td>
<td>89.2</td>
<td>69.4</td>
</tr>
</tbody>
</table>

Fig. 12. Average retrieval results from ImageCLEF for different methods.


