Abstract

Texture information plays an important role in image analysis. Although several descriptors have been proposed to extract and analyze texture, the development of automatic systems for image interpretation and object recognition is a difficult task due to the complex aspects of texture. Scale is an important information in texture analysis, since a same texture can be perceived as different texture patterns at distinct scales. Gray level co-occurrence matrices (GLCM) have been proved to be an effective texture descriptor. This paper presents a novel strategy for extending the GLCM to multiple scales through two different approaches, a Gaussian scale-space representation, which is constructed by smoothing the image with larger and larger low-pass filters producing a set of smoothed versions of the original image, and an image pyramid, which is defined by sampling the image both in space and scale. The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other well-known texture operators, including the original

*Corresponding author.
GLCM, that even though faster than the proposed method, is significantly outperformed in accuracy.

Keywords: Multi-scale feature descriptor, gray level co-occurrence matrix, GLCM, texture description, image analysis.

1. Introduction

Texture can be characterized by regular or random patterns that repeat over a region [1]. As one of the most important features for image analysis, texture provides information regarding structural arrangement of surfaces or changes in intensity or color brightness.

Despite the accuracy of the visual human system to recognize textures, it is a complex task to define a set of textural descriptors for image analysis on different domains of knowledge. The large number of definitions and descriptors found in the literature reflects such difficulty [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

Although there is no a unique categorization of the main relevant methods for texture description, they can be classified as statistical approaches, signal-processing based approaches, geometrical approaches, and parametric-model based approaches [13].

Among the statistical approaches, gray level co-occurrence matrices (GLCM) have been proved to be a very powerful texture descriptor used in image analysis. However, a drawback of the original GLCM is its limited capability of capturing texture information at multiple scales.

In many descriptors, it is assumed that texture information is fixed at a specific image resolution. The discriminative power of texture descriptors
can be significantly improved if different scales are considered among the images during the descriptor extraction.

This work presents a novel scheme for extending the GLCM to be more robust under scale variation. Two different multi-scale representations are used in the extension of the descriptor. The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other powerful texture operators.

The paper is organized as follows. Section 2 presents some relevant concepts and work related to texture descriptors. Section 3 describes the proposed multi-scale texture descriptor. Experimental results are shown in Section 4. Finally, Section 5 concludes the paper with final remarks.

2. Related Work

The development and analysis of low-level feature descriptors have been widely considered in the past years. Among the vastly employed methods are the scale-invariant feature transform (SIFT) [14], speeded up robust feature (SURF) [15], histogram of oriented gradients (HOG) [16], gradient location and orientation histogram (GLOH) [17], region covariance matrix (RCM) [18], edgelet [19], gray level co-occurrence matrix (GLCM) [20], local binary patterns (LBP) [21], color correlogram (CCG) [22], color coherence vectors (CCV) [23], color indexing [24], steerable filters [25], Gabor filters [26], and shape context [27]. Furthermore, several works comparing different feature descriptors can be found in the literature [17, 28, 29].

Some descriptors have taken scale changes into account. For instance,
extensions of LBP have been proposed to make it a more effective texture
descriptor. One extension is the multi-resolution LBP (MLBP) [30], which
uses pixel neighborhood of different sizes. Another extension represents LBP
descriptors in Gabor transform domain (LGBP) [31]. A third variation ex-
tends LBP to pyramid transform domain (PLBP) [32]. PHOG [33] represents
an image with histograms of orientation gradients over spatial pyramids.
HWVP [34] represents texture information of an image through a hierarchi-
cal wavelet packet transform. GIST [35] is a high dimensional descriptor
that represents texture information with filtering based on oriented multiple
scale.

Low-level descriptors are designed to focus in visual characteristics such as
texture, shape and color. Since this paper proposes a multi-scale extension
of the texture-based method GLCM, we review approaches to extracting
textural information. Section 2.1 describes a set of texture-based feature
descriptors. Then, Section 2.2 reviews the GLCM method and its extensions.

2.1. Texture Descriptors

Many approaches to extracting textural information have been proposed
in the literature, including gray level co-occurrence matrices [20], gray level
run length matrices [36], wavelet transforms [37], Gabor filters [38], texture
unit [39], local binary patterns [21], texture feature coding method [40], co-
ordinated clusters representation [41], granulometry [42], Markov random
fields [43], and simultaneous autoregressive models [44]. This section reviews
those approaches that have achieved accurate results in texture classifica-
tion [29, 45, 46, 47]. These descriptors are then compared to the proposed
method during the experimental validation (Section 4).
Granulometry. The term granulometry is used in the field of materials science to characterize the granularity of materials by passing them through sieves of different sizes while measuring their mass retained by each sieve.

This principle can be transposed to the field of image processing [42, 48, 49], where an operator consists in analyzing the amount of image detail removed by applying morphological openings $\gamma_\lambda$ of increasing size $\lambda$.

The mass is represented by the sum of the pixel values, known as image volume (Vol). The volumes of the opened images are plotted against $\lambda$, producing a granulometric curve. The normalized version of the operator for an image $f$ can be written as

$$G(f) = \frac{\text{Vol}(\gamma_\lambda(f))}{\text{Vol}(f)}$$  

(1)

Negative values of $\lambda$ can be interpreted as a morphological closing operator with a structuring element of size $\lambda$.

Local Binary Patterns. The Local Binary Patterns (LBP) descriptor characterizes the spatial structure of a texture and presents the characteristics of being invariant to monotonic transformations of the gray-levels [21]. On its standard version, a pixel $c$ with intensity $g(c)$ is labeled as defined by Equation 2, where pixels $p$ belong to a $3 \times 3$ neighborhood with gray levels $g_p$ ($p = 0, 1, 2, ..., 7$).

$$S(g_p - g_c) = \begin{cases} 
1, & g_p \geq g_c \\
0, & g_p < g_c
\end{cases}$$  

(2)

Then, the pattern of the pixel neighborhood for an image with 256 intensity values is computed by summing the corresponding thresholded values $S(g_p -$
\( g_c \) weighted by a binomial factor of \( 2^k \) as

\[
LBP = \sum_{p=0}^{7} S(g_p - g_c)2^p
\]  

(3)

After computing the labeling for each pixel of the image, a 256-bin histogram of the resulting labels is used as a feature descriptor for the texture.

Several variations of LBP have been proposed, including the Improved Local Binary Pattern (ILBP) [50]. Different from the LBP, the ILBP uses the average of the 3 \( \times \) 3 neighborhood as the threshold and also considers the central pixel to estimate its label, with values in the interval \([0, 510]\). The ILBP is defined as in Equation 4:

\[
ILBP = \sum_{p=0}^{7} S(g_p - m)2^p + S(g_c - m)2^8
\]  

(4)

where \( m \) denotes the average of the 3 \( \times \) 3 neighborhood and \( S(x) \) is defined as

\[
S(x) = \begin{cases} 
1, & x > 0 \\
0, & x \leq 0 
\end{cases}
\]  

(5)

\[
m = \frac{1}{9} \sum_{p=0}^{7} g_p + g_c
\]  

(6)

Similarly to LBP, once the labels have been computed for every pixel, a histogram is computed; in this case, a 511-bin histogram will be used as feature vector.

**Gabor Filters.** Gabor filters capture visual properties such as spatial localization, spatial frequency and orientation of the structures present in the image. Widely employed to object recognition, Gabor filters present illumination invariance since they detect invariant spatial frequency [38].
most common form of the Gabor filters is shown in Equation 7, where \( \mu \) and \( \nu \) denote the orientation and scale of the Gabor kernels, \( z = (x, y) \), \( \| \cdot \| \) is the norm operator, and \( k_{\mu,\nu} = k_\nu (\cos \phi_\mu, \sin \phi_\mu) \), in which \( k_\nu = k_{\text{max}}/f^\nu \) and \( \phi_\mu = \pi \mu/8 \) where \( k_{\text{max}} \) is the maximum frequency and \( f \) denotes the spacing factor between kernels in the frequency domain.

\[
\psi_{\mu,\nu}(z) = \frac{k_{\mu,\nu}}{\sigma^2} e^{-((k_{\mu,\nu})^2/2\sigma^2)} \left[ e^{ik_{\mu,\nu}z} - e^{-\sigma^2/2} \right] (7)
\]

The feature vector extracted using the Gabor filters is obtained with the convolution of the gray-scale image with the filters. Let \( I(x, y) \) be the image, its convolution with a Gabor filter is defined according to Equation 8, where \( * \) denotes the convolution operator.

\[
G_{\psi I}(x, y, \mu, \nu) = I(x, y) * \psi_{\mu,\nu}(z) (8)
\]

In this work, we consider five scales \( \mu \in \{0, \ldots, 4\} \) and eight orientations \( \nu \in \{0, \ldots, 7\} \), which result in 40 Gabor filters. All filters are convolved and the resulting magnitudes are used as descriptors. After the convolution, the feature vector is composed by the concatenation of the mean and standard deviation of the convolved image.

2.2. Gray Level Co-occurrence Matrix

An approach to extracting textural information regarding gray level transition between two pixels uses a co-occurrence matrix. Given a spatial relationship defined among pixels in a texture, such matrix represents the joint distribution of gray-level pairs of neighboring pixels. Therefore, matrices providing different information are obtained by modifying the spatial relationship (different orientation or distance between pixels). Descriptors are extracted from these matrices.
The number of rows and columns of the co-occurrence matrix depends only on the gray levels in the texture and not on the image size. The element $P(m,n)$ of a co-occurrence matrix indicates the number of transitions between the gray level $m$ and $n$ that take place in the texture according to a given spatial relationship. It is customary to quantize the number of gray level intensities to reduce the size of the co-occurrence matrix. The number of bins usually varies from 8 to 256.

Before computing the co-occurrence matrix, it is necessary to define relations among pixels, that is, the arrangement of pixels from which the transitions will be considered. A set $S$ is built. Each element in this set is a pair of coordinates of each pixel involved in the relationship. Once $S$ is defined, Equation 9 is used to count the number of transitions between each pair of gray levels in the texture. In this equation, $f(x,y)$ indicates the gray level of a pixel located at $(x,y)$ in the image.

$$P(m,n) = \# \{(i,j),(k,l) \in S \mid f(i,j) = m \text{ and } f(k,l) = n\}$$ \quad (9)

Once the frequency of each gray level transition is computed, $P(m,n)$ is placed at the $m$-th row and $n$-th column of the matrix. Then, feature descriptors are extracted after a normalization based on Equation 10, where $H_g$ denotes the largest gray level.

$$p_{m,n} = \frac{P(m,n)}{\sum_{i=0}^{H_g} \sum_{j=0}^{H_g} P(i,j)} \quad m, n = 0, \ldots, H_g$$ \quad (10)

According to Equation 9, the co-occurrence matrix depends on the gray level transitions between pairs of pixels in set $S$. This way, it is possible to
arbitrarily specify the distance and the angle between the pairs. Haralick et al. [20] defined specifically which transitions should be considered to compute co-occurrence matrices. Two additional parameters are included, $d$ and $\theta$. These parameters define the displacement and angle between pixels in $S$, respectively. Therefore, several matrices can be obtained with small changes in these parameters.

To describe the properties contained in the co-occurrence matrices, Haralick et al. proposed 14 statistical measures that are computed from the matrices: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, two information measures of correlation, and maximal correlation coefficient.

Besides the original version of the GLCM, several variations have been proposed. Focusing on optimization, Clausi and Jernigan [51] employ linked lists exploiting the sparsity of the co-occurrence matrices to reduce the computation time; Tahir et al. [52] propose the use of Field Programmable Gate Arrays (FPGA) to accelerate the calculation of GLCM and Haralick texture features, achieving speed-up of 9 times.

Extensions of the GLCM have been proposed. To increase the discriminability of the descriptors, Gelzinis et al. [53] extract descriptors considering simultaneously different values for parameter $d$. Walker et al. [54] have proposed to form co-occurrence matrix-based features by weighted summation of GLCM elements from areas presenting high discrimination. Furthermore, addition of color information has also been considered for co-occurrence matrices [55].
Multi-scale analysis has also been performed using the GLCM. Hu [56] and Pacifici et al. [57] consider multiple scales by changing the window size from which the GLCM descriptors are extracted. Rakwatin et al. [58] propose that the image be rescaled to different sizes, extracting co-occurrence descriptors from each size. Nguyen-Duc et al. [59] have obtained improved results on content-based image retrieval employing a combination of contourlet transform [60] and GLCM. First, the contourlet transform is performed for four subbands of the image, then the GLCM features are extracted from each one.

Differently from previous works that have addressed multiple scales for GLCM, this paper exploits two approaches to performing the multi-scale analysis on the images: pyramid decomposition and Gaussian smoothing. Furthermore, five strategies to combine the descriptors extracted from each scale are considered and evaluated.

3. Multi-Scale Gray Level Co-Occurrence Matrix

Two common multi-resolution representations are Gaussian smoothing and pyramid decomposition [61]. Gaussian smoothing is a space-scale representation, where a sequence of images at different levels of space-scale are built with variable kernel sizes. In pyramid representation, the original image is progressively reduced at each level of the pyramid. Pyramid images can be generated by applying the Gaussian smooth filtering, Laplacian operator, low-pass filters of wavelet transform, among other schemes.

The use of multiple scales can improve the discriminative power of texture descriptors, since they are able to extract information that could not be
present in certain scales. This work proposes an extension of the gray level co-
ocurrence matrix (GLCM) to multiple scales using the Gaussian smoothing
and pyramid decomposition to generate a number of image representations
of the original image.

The co-occurrence matrix is created at each level of resolution, resulting
in several scales. Features extracted from GLCM are then combined into a
single feature descriptor. All features are extracted from images in grayscale,
that is, color data sets are converted into grayscale before their extraction.

Figure 1 illustrates the main steps of the multi-scale approaches.

![Figure 1: Multi-scale approaches for GLCM. (a) Pyramid decomposition; (b) Gaussian smoothing.](image)

Once the features are extracted from each scale, it is necessary to combine
them in a manner to take advantage of the multiple scale information. To
achieve that, five different combination strategies are proposed to merge the
features.

The first one, referred to as *concatenation*, is a simple union of the scales,
that is, features extracted from each scale are concatenated in a single feature
vector that contains all the information.

In the second method, referred to as *local*, a normalization is applied to
each feature descriptor extracted from a certain scale, then all normalized
features are joined together through a simple concatenation.

The third scheme, *(L + G)*, is achieved by applying a normalization
to the feature descriptors extracted from each scale, followed by a global
normalization of all features united in a single feature vector.

The fourth strategy, referred to as *corresponding*, normalizes correspond-
ing elements of the scales in the descriptor, that is, the first feature extracted
from the first scale is normalized in relation to the first features of every other
scale, then the second feature is normalized according to the second feature
of each scale, and so forth. This combination is adequate when features have
very discrepant value intervals, even in a single scale.

Finally, since all the previous multi-scale combination strategies increase
the size of the final feature descriptor, a fifth method, referred to as *weighted*,
is proposed to maintain the size of the descriptor equal to the number of fea-
tures extracted from a single scale. Instead of a normalization, as used in
the fourth method, a weighted mean is used to combine the values of corre-
sponding elements of each scale. Each scale will be assigned with decreasing
weight of value $1/2^k$, where $k$ is the number of the current scale.
4. Experimental Results

This section describes and discusses the results obtained through the evaluation of the feature descriptors applied to texture classification. All experiments were conducted on an Intel Core i7-2630QM processor, 2.2 GHz with 8 GB of RAM running 64-bit Windows operating system. The method was implemented using C++ programming.

We consider five texture data sets: UIUC [62], UMD [63], Brodatz [64], OuTex [65] and VisTex [66]. All data sets were used to estimate and evaluate the parameters employed to assess the effectiveness of the proposed multi-scale feature descriptor for texture classification.

Section 4.1 presents a brief explanation of each data set used in our experiments and the classification protocols employed. In Section 4.2, we describe the parameter values used for each method. Then, evaluations of the proposed approach and comparisons to other feature extraction methods are presented and discussed in Section 4.3.

4.1. Data Sets

This section describes the main characteristics of the five data sets used in our experiments.

4.1.1. UMD Data Set

The UMD high-resolution data set [63] contains images of $1280 \times 960$ pixels, with 1000 images split into 25 classes, giving a total of 40 samples per class. This data set includes images of floor textures, plants, fruits, among others. A mosaic containing examples of all classes can be seen in Figure 2.
4.1.2. UIUC Data Set

The UIUC data set [62] is a texture data set composed of 1000 images of $640 \times 480$ pixels, distributed in 25 classes (variations of wood, gravel, fur, carpet, brick, among others) with 40 samples each. Figure 3 shows examples of images for each texture.

4.1.3. OuTex Data Set

OuTex [67] is a framework for evaluation of texture classification and segmentation. It contains several images and protocols for texture classification. Different textures were acquired under several illuminations and different orientations.

Several test suites for texture classification have been proposed. Out of
Figure 3: Examples of 25 texture samples extracted from UIUC data set [62].

all these, TC_00005 is used in this work since its results are not saturated.  
This test suite contains 8832 sample images of 32 × 32 pixels, belonging to  
24 classes of textures. The classification protocol for this suite is composed of  
one hundred different combinations for training and test. We execute for all  
of them and report the average classification rates. Figure 4 shows examples  
of texture images.

4.1.4. VisTex Data Set

VisTex [66] is a collection of texture images that are representative of  
real world conditions. Our experiments included 54 images of resolution  
with 512 × 512 pixels split into 16 samples of 128 × 128 pixels, according to  
work described by Arvis et al. [68]. Such images are available at the OuTex
site [67] as test suite Contrib_TC_00006. According to this suite, for each texture class, half of the samples are used in the training set and the other half are used as test data. Figure 5 shows examples of texture images.

4.1.5. Brodatz Data Set

The Brodatz [64] photo album is a widely used texture data set, often treated as a benchmark for texture classification. It contains 111 different texture classes with 512 × 512 pixels, which are subdivided to compose several classes. The test suite Contrib_TC_00004 is used in this work (available at the OuTex site [67]). It is composed of 2048 images of 64 × 64 pixels divided equally among 32 classes. Ten combinations for training and test are considered for this suite. Figure 6 shows a mosaic containing samples of Brodatz data set.
Figure 5: Examples of texture samples extracted from Vistex data set [66].

4.2. Experimental Setup

For all experiments, the nearest neighbor classifier is applied after all variables are normalized to present zero mean and unit variance. In addition, Principal Component Analysis (PCA) is applied to avoid dimensionality issues – the estimation of the best dimensionality is performed by cross-validation and it is allowed to vary between 5 and 40 dimensions. The dimensionality which achieved the best classification rate for the data set was determined and chosen, whose values are presented in Table 4.

A subset of 12 descriptors from the original 14 descriptors described in
Section 2.2 is considered for the GLCM: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy and maximal correlation coefficient. Co-occurrence matrices for four orientations are computed (0°, 45°, 90° and 135°) and used to compose the feature vector with 48 dimensions (12 per orientation).

The values for parameters $d$ and the number of bins for the GLCM are experimentally estimated to be 1 and 256, respectively (these values are used throughout the experiments). Table 1 shows the parameter estimation.
According to the table, the higher the number of bins, the better are the results for most data sets. However, the displacement parameter \( d \) has not presented the same behavior. For data sets with larger samples such as UMD and UIUC, larger values of the displacement provide better results, while the opposite behavior takes place in data sets with samples with small size (VisTex and OuTex).

Regarding the remaining feature descriptors that will be considered in the comparison, for the granulometry we used kernel size equal to 25 and step equal to 2 in UIUC and UMD data sets, and kernel size equal to 5 and step equal to 1 for the remaining data sets, which leads to feature vectors with 104 and 40 dimensions, respectively. In Gabor filters, we used 8 orientations and 5 scales, generating 40 filters for convolution. For each filtered image, two measures were extracted (mean and standard deviation), which lead to 80 descriptors. The LBP and ILBP were used with their standard configuration and the resulting feature vectors have 256 and 512 dimensions, respectively.

4.3. Evaluations and Comparisons

We have evaluated the number of decomposition levels, approaches used to decompose the images into multiple scales, and strategies for combining the feature descriptors extracted from different scales. The results are shown in Tables 2 and 3. According to them, the scale decomposition based on Gaussian smoothing has provided higher classification rates for all data sets and the most significant difference occurs with the OuTex data set (84.61% with Pyramid decomposition and 89.50% with Gaussian smoothing). This is because the small sample size for this data set associated with the pyramid decomposition results in the computation of the GLCM for very small
images, resulting, therefore, on sparse matrices. The Gaussian smoothing decomposition is considered during the comparison to other methods.

Tables 2 and 3 also evaluate five approaches to combine the descriptors extracted from each scale. According to the results, three approaches have achieved higher classification rates: local normalization, normalization based on corresponding descriptors, and simple concatenation without normalization. The combination strategy, based on reducing the weight of the descriptors extracted from smaller scales, showed a weak performance (meaning that the importance of all scales should be equally considered). The combination strategy that achieved the best results for each data set is used in the comparisons to other methods.

Finally, comparisons among the proposed multi-scale feature descriptor with other well-known methods for texture analysis are shown in Table 4. For the UMD and UIUC data sets, we show the results when a different number of samples are used for training (as indicated in the second column). The remaining datasets follow the protocols described in Section 4.1.

There are important observations that can be made according to the results shown in Table 4. First, significant improvements have been achieved for all data sets when the multi-scale GLCM is employed, compared to the original approach. Second, considering the other methods in the literature, the proposed approach achieved the best classification rates on three out of the five tested data sets. Furthermore, it is also important to point out that the most significant improvements were achieved on data sets presenting samples with large sizes (UMD and UIUC), in which more information can be captured by a multi-scale approach. Finally, although the Gabor filters
Table 1: Parameter estimation for all data sets considering single scale GLCM (20 training samples were considered for UMD and UIUC data sets). The results show classification rates, in percentages.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>d</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMD</td>
<td>1</td>
<td>90.07</td>
<td>90.86</td>
<td>91.60</td>
<td>91.62</td>
<td>91.29</td>
<td>91.56</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>91.49</td>
<td>92.36</td>
<td>92.66</td>
<td>92.44</td>
<td>92.17</td>
<td>92.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>91.90</td>
<td>92.60</td>
<td>92.68</td>
<td>92.52</td>
<td>92.59</td>
<td><strong>92.78</strong></td>
</tr>
<tr>
<td>UIUC</td>
<td>1</td>
<td>65.15</td>
<td>67.23</td>
<td>68.23</td>
<td>68.21</td>
<td>68.83</td>
<td>71.38</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>71.44</td>
<td>72.47</td>
<td>71.55</td>
<td>72.26</td>
<td>72.27</td>
<td>73.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>73.70</td>
<td><strong>75.67</strong></td>
<td>75.56</td>
<td>75.06</td>
<td>74.16</td>
<td>75.53</td>
</tr>
<tr>
<td>Brodatz</td>
<td>1</td>
<td>67.06</td>
<td>70.46</td>
<td>72.27</td>
<td>74.24</td>
<td>75.63</td>
<td>76.28</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>73.37</td>
<td>76.69</td>
<td>78.30</td>
<td><strong>78.58</strong></td>
<td>77.97</td>
<td>77.80</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>65.30</td>
<td>70.31</td>
<td>72.68</td>
<td>76.13</td>
<td>74.59</td>
<td>73.83</td>
</tr>
<tr>
<td>VisTex</td>
<td>1</td>
<td>93.75</td>
<td>93.98</td>
<td>93.52</td>
<td>93.75</td>
<td>93.98</td>
<td>93.98</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>93.98</td>
<td>93.52</td>
<td>92.82</td>
<td>92.82</td>
<td>93.52</td>
<td>93.29</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>90.74</td>
<td>90.97</td>
<td>90.51</td>
<td>90.97</td>
<td>91.20</td>
<td>91.20</td>
</tr>
<tr>
<td>OuTex</td>
<td>1</td>
<td>77.56</td>
<td>78.39</td>
<td>81.18</td>
<td>80.98</td>
<td>82.22</td>
<td><strong>83.02</strong></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>78.56</td>
<td>78.13</td>
<td>79.83</td>
<td>79.92</td>
<td>80.75</td>
<td>82.19</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>77.08</td>
<td>75.61</td>
<td>77.07</td>
<td>76.86</td>
<td>78.35</td>
<td>80.00</td>
</tr>
</tbody>
</table>

also perform multi-scale analysis, the classification rates are not as high as the ones achieved by the proposed approach.

Table 5 shows the computational time required for each feature in all tested data sets. It can be observed that multi-scale GLCM takes more time than its single-scale version, which is expected since that the GLCM has to be computed in every level of the pyramid in the proposed method.
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Scales</th>
<th>Combination Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local</td>
</tr>
<tr>
<td>UMD</td>
<td>2</td>
<td>90.84</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>88.26</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>89.88</td>
</tr>
<tr>
<td>UIUC</td>
<td>2</td>
<td>71.82</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>72.18</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>73.19</td>
</tr>
<tr>
<td>Brodatz</td>
<td>2</td>
<td>68.27</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>65.77</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>58.13</td>
</tr>
<tr>
<td>VisTex</td>
<td>2</td>
<td>90.28</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>90.05</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>87.04</td>
</tr>
<tr>
<td>OuTex</td>
<td>2</td>
<td>84.61</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>82.28</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>76.66</td>
</tr>
</tbody>
</table>

Table 2: Classification rates (%) achieved when the pyramid decomposition is considered for the GLCM and multiple approaches for feature descriptor combination are employed, as described in Section 3 (due to limited space, we used the term L+G, which means local followed by global normalization).

5. Conclusions

This paper describes an extension of the GLCM texture descriptor to multiple scales based on a Gaussian smoothing approach and a pyramid decomposition. Features extracted from GLCM at different scales are combined and evaluated for each multi-resolution representation.
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Scales</th>
<th>Combination Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local</td>
</tr>
<tr>
<td>UMD</td>
<td>2</td>
<td>92.40</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>88.25</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>88.46</td>
</tr>
<tr>
<td>UIUC</td>
<td>2</td>
<td>80.08</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>80.13</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>81.88</td>
</tr>
<tr>
<td>Brodatz</td>
<td>2</td>
<td>80.82</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>81.21</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>81.29</td>
</tr>
<tr>
<td>VisTex</td>
<td>2</td>
<td>94.44</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>91.90</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>91.90</td>
</tr>
<tr>
<td>OuTex</td>
<td>2</td>
<td>88.71</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td><strong>89.50</strong></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>89.33</td>
</tr>
</tbody>
</table>

Table 3: Classification rates (%) achieve when the Gaussian smoothing is considered for the GLCM and multiple approaches for feature descriptor combination are employed, as described in Section 3 (due to limited space, we used the term L+G, which means local followed by global normalization).

The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other well-known descriptors for texture analysis. The proposed multi-scale GLCM achieved significant improvements for all tested data sets.
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Feature Extraction Method</th>
<th>LBP</th>
<th>ILBP</th>
<th>Gabor</th>
<th>Granulometry</th>
<th>GLCM</th>
<th>GLCM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMD</td>
<td></td>
<td>72.5</td>
<td>73.3</td>
<td>75.7</td>
<td>75.4</td>
<td>83.1</td>
<td>85.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>81.5</td>
<td>81.3</td>
<td>82.3</td>
<td>83.8</td>
<td>88.7</td>
<td>91.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.4</td>
<td>85.0</td>
<td>84.9</td>
<td>88.0</td>
<td>91.1</td>
<td>93.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>88.1</td>
<td>87.4</td>
<td>88.1</td>
<td>90.6</td>
<td>92.7</td>
<td>95.2</td>
</tr>
<tr>
<td>UIUC</td>
<td></td>
<td>41.7</td>
<td>44.8</td>
<td>51.0</td>
<td>48.3</td>
<td>58.9</td>
<td>65.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>44.6</td>
<td>51.7</td>
<td>59.4</td>
<td>58.9</td>
<td>68.5</td>
<td>74.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>49.2</td>
<td>54.7</td>
<td>65.1</td>
<td>64.5</td>
<td>72.7</td>
<td>78.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>53.4</td>
<td>61.3</td>
<td>69.9</td>
<td>68.6</td>
<td>75.6</td>
<td>81.7</td>
</tr>
<tr>
<td>Brodatz</td>
<td></td>
<td>87.1</td>
<td></td>
<td>89.4</td>
<td>89.4</td>
<td>88.8</td>
<td>78.3</td>
</tr>
<tr>
<td>VisTex</td>
<td></td>
<td>96.5</td>
<td></td>
<td>97.9</td>
<td>92.6</td>
<td>94.4</td>
<td>96.0</td>
</tr>
<tr>
<td>OuTex</td>
<td></td>
<td>89.0</td>
<td></td>
<td>82.0</td>
<td>84.9</td>
<td>89.2</td>
<td>83.0</td>
</tr>
</tbody>
</table>

Table 4: Texture classification results achieved by multiple texture descriptors for different data sets. The second column indicates the number of samples used during training for the UMD and UIUC data sets. The results show classification rates on percentage.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Feature Extraction Method</th>
<th>LBP</th>
<th>ILBP</th>
<th>Gabor</th>
<th>Granulometry</th>
<th>GLCM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMD</td>
<td></td>
<td>94.86</td>
<td>105.85</td>
<td>4568.99</td>
<td>9979.91</td>
<td>135.84</td>
</tr>
<tr>
<td>UIUC</td>
<td></td>
<td>24.95</td>
<td>27.21</td>
<td>1371.59</td>
<td>2536.75</td>
<td>39.35</td>
</tr>
<tr>
<td>Brodatz</td>
<td></td>
<td>0.78</td>
<td>0.88</td>
<td>110.12</td>
<td>68.51</td>
<td>12.30</td>
</tr>
<tr>
<td>VisTex</td>
<td></td>
<td>1.95</td>
<td>2.14</td>
<td>95.17</td>
<td>116.60</td>
<td>8.05</td>
</tr>
<tr>
<td>OuTex</td>
<td></td>
<td>1.89</td>
<td>3.96</td>
<td>297.94</td>
<td>79.20</td>
<td>41.66</td>
</tr>
</tbody>
</table>

Table 5: Computational time (in seconds) required to perform the feature extraction.
References


[38] Daugman, J.G.. Uncertainty Relation for Resolution in Space, Spatial
Frequency, and Orientation Optimized by Two-Dimensional Visual Cor-
1169.

[39] He, D.C., Wang, L.. Texture Unit, Texture Spectrum, and Tex-
ture Analysis. IEEE Transactions on Geoscience and Remote Sensing

[40] Horng, M.H., Sunb, Y.N., Lin, X.Z.. Texture Feature Coding Method
for Classification of Liver Sonography. Computerized Medical Imaging


Morphological Texture Classification. In: et al., C.R., editor. Mathem-
377–386.


Multiresolution Simultaneous Autoregressive Models. Pattern Recogni-


