

Evaluation of Feature Descriptors for Texture Classification

William Robson Schwartz^{1,2}, Fernando Roberti de Siqueira²,

Helio Pedrini²

¹Dept. of Computer Science, Federal University of Minas Gerais

Belo Horizonte-MG, Brazil, 31270-010

²Institute of Computing, University of Campinas

Campinas-SP, Brazil, 13083-852

ABSTRACT

Successful execution of tasks such as image classification, object detection and recognition, and scene classification depends on the definition of a set of features able to describe images effectively. Texture is among the features used by the human visual system. It provides information regarding spatial distribution, changes in brightness, and description regarding the structural arrangement of surfaces. However, although the visual human system is extremely accurate to recognize and describe textures, it is difficult to define a set of textural descriptors to be used in image analysis on

Further author information: (Send correspondence to W.R.S.)

W.R.S.: E-mail: william@dcc.ufmg.br, Telephone: +55 31 3409-5847

F.S.: E-mail: fernandosiq@gmail.com

H.P.: E-mail: helio@ic.unicamp.br, Telephone: +55 19 3521-5919

different application domains. This work evaluates several texture descriptors and demonstrates that the combination of descriptors can improve the performance of texture classification.

Keywords: Feature descriptors, texture analysis, feature fusion, texture classification

1. INTRODUCTION

The definition of a set of visual features able to describe images effectively, so that classification, detection and recognition processes can be applied, is a complex task in image analysis. A way to address this problem is to recur to features used by humans to understand visual information.

Texture is among the features used by the human visual system and can be characterized by local variations of pixel values that repeat in a regular or random pattern on the object or image. It can also be defined as a repetitive arrangement of patterns over a region.¹ It provides information regarding spatial distribution, changes in brightness, and description regarding the structural arrangement of surfaces. Therefore, the use of textural features is an important source of information for image description.

Although the visual human system is extremely accurate to recognize and describe textures, it is difficult to define a set of textural descriptors to be used in image analysis on different application domains, or even to formalize a definition for texture. Such difficulty is reflected by the large number of definitions and descriptors found in the literature.²⁻⁷

Feature descriptors are extracted from the input image and can be based on second-order statistics, parametric models, coefficients obtained from an image transform, or even a combination of these measures.

Texture classification usually involves two main stages, the learning step and the recognition step. In the first stage, a model is built to represent the texture content of each class present in the training data. In the second stage, the texture content of an unknown sample is extracted and compared to those extracted in the learning step. The sample is labeled to the class with the best match.

This work describes and compares a large set of feature descriptors in order to assess which are more suitable to be applied to texture classification. Furthermore, an experimental evaluation is presented to demonstrate that the combination of features produces superior results in terms of classification rate when compared to the individual use of the features, which means that some of them are complementary.

The paper is organized as follows. Section 2 presents texture descriptors considered in the evaluation. Experimental results are shown and discussed in Section 3. Finally, Section 4 concludes the paper with final remarks.

2. TEXTURE DESCRIPTORS

This section describes a number of relevant methods for texture feature extraction. Even though there is no a unique taxonomy to classify such methods, this work categorizes them into the following categories: *statistical approach* (Section 2.1), *approach based on signal processing* (Section 2.2), *geometrical approach* (Section 2.3),

and *approach based on parametric models* (Section 2.4). This taxonomy is based on that proposed by Tuceryan and Jain.⁸

2.1 Statistical Approach

Methods based on the statistical approach do not explore hierarchical structures presented by the texture, but represent its properties in indirect and probabilistic manners.

The simplest primitive that can be defined in a grayscale digital image is the pixel, which has the gray level as property. Consequently, the gray level distribution could be described by first order statistics, such as mean and variance estimated from a histogram computed from this distribution. However, since the first order statistics consider only pixels individually, this makes such measures more sensitive to changes in the image. Therefore, to avoid this problem, second order statistics, which depend on transitions between gray level of pixels, are considered.

The following sections describe the main statistical methods for texture analysis, including those based on first order statistics, co-occurrence matrix, features extracted from higher order statistics, such as gray level run length matrices, and autocorrelation function.

2.1.1 First Order Statistics

From the gray level histogram of a textured image, it is possible to extract first order statistics. Given an image with n pixels, the histogram can be computed using

Equation 1, where $h(i)$ represents the number of occurrences of the i -th gray level.

$$P(i) = \frac{h(i)}{n} \quad (1)$$

Although first order statistics present disadvantages, the computational cost to extract descriptors is very low since only simple measures, such as mean, variance, skewness, and kurtosis, need to be computed. These measures are shown in Equations 2 to 5, respectively, where H_g denotes the largest gray level in the image.

$$\mu = \frac{1}{n} \sum_{i=0}^{H_g} h(i) \quad (2)$$

$$\sigma^2 = \frac{1}{n-1} \sum_{i=0}^{H_g} (h(i) - \mu)^2 \quad (3)$$

$$s = \frac{1}{n\sigma^3} \sum_{i=0}^{H_g} (h(i) - \mu)^3 \quad (4)$$

$$k = \frac{1}{n\sigma^4} \sum_{i=0}^{H_g} (h(i) - \mu)^4 - 3 \quad (5)$$

Energy (Equation 6) and entropy (Equation 7) are also other measures computed from the image histogram.

$$E = \sum_{i=0}^{H_g} (P(i))^2 \quad (6)$$

$$H = - \sum_{i=0}^{H_g} P(i) \log(P(i)) \quad (7)$$

2.1.2 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.

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 8 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.

$$P(m, n) = \#\{(i, j), (k, l) \in S \mid f(i, j) = m \text{ and } f(k, l) = n\} \quad (8)$$

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 9, where H_g denotes the largest gray level.

$$p_{m,n} = \frac{P(m,n)}{\sum_{i=0}^{H_g} \sum_{j=0}^{H_g} P(i,j)} \quad m, n = 0, \dots, H_g \quad (9)$$

According to Equation 8, 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.⁹ defined specifically which transitions should be considered to compute co-occurrence matrices. Two additional parameters are included, d and θ . These parameters define the distance and angle between pixels in S , respectively. Therefore, several matrices may 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.

2.1.3 Gray Level Run Length Matrices

Gray level runs are obtained by sampling co-linear regions with the same gray level in an image. Aiming at summarizing the information obtained by the runs, Galloway¹⁰ proposed a matrix to tabulate the number of runs with specific lengths for given gray levels. Consequently, high order statistics computed from this matrix can be used for texture analysis.

From these matrices, called gray level run length matrices (GLRLM), relevant information regarding the texture can be extracted. It is expected long runs to be more frequent in coarse textures, while short runs are expected in fine textures due to edges and abrupt changes in the gray level.

The matrices proposed by Galloway are built as follows. Given a fixed orientation θ , the set composed by consecutive pixels with the same gray level on this orientation represents a gray level run and the number of pixels in this run is denoted by run length. Then, each entry of the GLRLM, $P(i, j|\theta)$, contains the number of runs with length j with gray level i for an orientation θ . Although the orientation θ can assume any value, in general the GLRLM are computed for a subset of orientations: $0^\circ, 45^\circ, 90^\circ$ and 135° . Figure 1 illustrates the runs for these orientations.

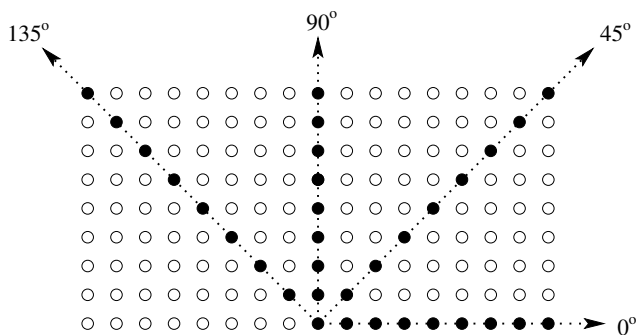


Figure 1. Orientations from which the GLRLM are usually computed.

Once the GLRLM are available, measures are computed and used as descriptors. The measures proposed by Galloway are described as follows. In these equations, H_g and N_r denote the largest gray level and the largest run length, respectively.

Short runs emphasis (SRE) and long runs emphasis (LRE), shown in Equations 10

and 11. LRE presents large values when large run lengths are present in the texture.

$$\text{SRE} = \frac{\sum_{i=0}^{H_g} \sum_{j=1}^{N_r} \frac{P(i, j|\theta)}{j^2}}{\sum_{i=0}^{H_g} \sum_{j=1}^{N_r} P(i, j|\theta)} \quad (10)$$

$$\text{LRE} = \frac{\sum_{i=0}^{H_g} \sum_{j=1}^{N_r} j^2 P(i, j|\theta)}{\sum_{i=0}^{H_g} \sum_{j=1}^{N_r} P(i, j|\theta)} \quad (11)$$

Measures of gray level non-uniformity (GLN) and run length non-uniformity (RLN), shown in Equations 12 and 13, respectively, are used to describe the distribution of gray levels and run lengths. For example, GLN presents small values when the number of runs is uniformly distributed according to the gray level.

$$\text{GLN} = \frac{\sum_{i=0}^{H_g} \left(\sum_{j=1}^{N_r} P(i, j|\theta) \right)^2}{\sum_{i=0}^{H_g} \sum_{j=1}^{N_r} P(i, j|\theta)} \quad (12)$$

$$\text{RLN} = \frac{\sum_{i=1}^{N_r} \left(\sum_{j=0}^{H_g} P(i, j|\theta) \right)^2}{\sum_{i=0}^{H_g} \sum_{j=1}^{N_r} P(i, j|\theta)} \quad (13)$$

The last measure proposed by Galloway, called run percentage (RP), is shown in Equation 14, where n denotes the number of pixels in the image. This measure presents large values when the texture is mostly composed of short runs.

$$\text{RP} = \frac{1}{n} \sum_{i=0}^{H_g} \sum_{j=1}^{N_r} P(i, j|\theta) \quad (14)$$

2.1.4 Autocorrelation Function

An approach to discriminating between coarse and fine textures is based on spatial frequency. Fine textures are composed of small primitives and therefore present high

spatial frequency due to the large number of gray level variations, whereas coarse textures possess large primitives with low spatial frequency.

The autocorrelation function describes spatial interactions between the primitives composing a texture.² In this case, the gray levels are assumed to be the primitives that compose a texture, while the interactions among these primitives are characterized by the autocorrelation coefficient. This coefficient is obtained by Equation 15 for a texture with $M \times N$ pixels. For each pair of values $\{p, q\}$, the texture is shifted by at most p pixels in x coordinate and q pixels in y coordinate.

$$\rho_{ff}(p, q) = \frac{\sum_{i=0}^{M-p-1} \sum_{j=0}^{N-q-1} f(i, j)f(i + p, j + q)}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f^2(i, j)} \quad (15)$$

2.2 Approach Based on Signal Processing

Texture analysis methods based on signal processing extract descriptors from an image representation obtained by applying image transforms, such as Fourier or wavelet transforms.

2.2.1 Fourier Spectrum

The spectrum resulting from the 2D Fourier transform (shown in Equation 16 for a $n \times n$ image, where $i = \sqrt{-1}$), after shifting the frequency plane from the origin, shows high energy concentration at the center when the image presents low spatial frequency, whereas the energy is more spread when the image has high frequency.

Extending this concept to texture analysis, coarse textures will present high concentration of energy at the center due to homogeneity. On the other hand, the energy will be spread around the plane when fine textures are considered.

$$\mathcal{F}(u, v) = \frac{1}{n^2} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} f(k, l) \exp(-2\pi i(ku + lv)/n) \quad (16)$$

Converting the Fourier spectrum, S , to a polar representation $S(r, \theta)$, r and θ are the variables in this coordinate system. For each direction θ , $S(r, \theta)$ can be considered as a function $S_\theta(r)$ and, similarly, for each radius r , $S_r(\theta)$ is also a unidimensional function. Therefore, the analysis of $S_\theta(r)$ for a given value of θ gives the behavior of the spectrum along with a radial direction and the analysis of $S_r(\theta)$ for a given value of r explains the behavior of the spectrum along a circle centered at the origin.

A global description is given by functions shown in Equations 17 and 18. Unidimensional descriptions of the spectrum are obtained by changing values of pairs $\{S(r), S(\theta)\}$. These values are used as texture descriptors.

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \quad (17)$$

$$S(\theta) = \sum_{r=1}^R S_r(\theta) \quad (18)$$

2.2.2 Wavelet Transforms

Wavelet transforms decompose a signal by means of a series of elementary functions, created from dilations and translations of a basis function ψ , known as mother

wavelet. The basis functions of a discrete wavelet transform, $\psi_{j,k}(t)$, of time independent variable t , can be expressed as

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) \quad (19)$$

where j and k are integers that guide the dilations and translations of the function ψ to generate a family of wavelets, such as Haar and Daubechies.^{11,12}

Wavelet transforms provide simultaneous time and frequency localization, whereas the standard Fourier transform is only localized in frequency. Additionally, wavelet transforms are useful for analyzing time-variant, non-stationary signals.

Using wavelets as a set of basis functions, an image can be decomposed into a multi-resolution hierarchy of localized information at different frequencies. The use of wavelet transform for texture analysis was first proposed by Mallat.¹²

Wavelet transforms can be implemented by using a pair of low-pass and high-pass filters represented by a sequence of coefficients. In a 2D wavelet decomposition, the filters are applied to an image in both horizontal and vertical directions, typically followed by a downsampling. The output of each level will generate four subband images, LL, LH, HL and HH. The same process can be repeated on the LL image to generate the next decomposition level.

As wavelet coefficients in different frequency bands show variations in horizontal, vertical and diagonal directions, it has been shown that texture features can be extracted from these coefficients.¹³

A well known feature based on wavelet coefficients is the energy, shown in Equa-

tion 20, where sb denotes the LL, LH, HL and HH subbands, $c(x, y)$ represents wavelet transform coefficients in the coordinates (x, y) for each one of these subbands containing $m \times m$ pixels. Wavelet energy reflects the distribution of energy along the frequency axis over scale and orientation and have proven to be very useful for texture characterization.

$$E_{sb} = \sqrt{\frac{1}{m^2} \sum c(x, y)^2} \quad (20)$$

2.2.3 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.¹⁴ The most common form of the Gabor filters is shown in Equation 21, where μ and ν 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 $k_{\nu} = k_{max}/f^{\nu}$ and $\phi_{\mu} = \pi\mu/8$ where k_{max} it the maximum frequency and f denotes the spacing factor between kernels in the frequency domain.

$$\psi_{\mu, \nu}(z) = \frac{\|k_{\mu, \nu}\|^2}{\sigma^2} e^{(-\|k_{\mu, \nu}\|^2 \|z\|^2 / 2\sigma^2)} [e^{ik_{\mu, \nu}z} - e^{-\sigma^2/2}] \quad (21)$$

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 22, where $*$ denotes the convo-

lution operator.

$$G_{\psi I}(x, y, \mu, \nu) = I(x, y) * \psi_{\mu\nu}(z) \quad (22)$$

In this work, we consider five scales $\mu \in \{0, \dots, 4\}$ and eight orientations $\nu \in \{0, \dots, 7\}$, which results in 40 Gabor filters. For each filter, the image is convolved, generating 40 magnitudes. The mean and standard deviation of each image are calculated, resulting in 80 different features for the Gabor filters.

2.3 Geometrical Approach

In the geometrical approach, a texture is defined as being composed of primitives, also known as textels. After identifying primitives composing the texture, two classes of methods can be considered for feature extraction. The first uses descriptors extracted from the primitives to describe the texture, while the second considers rules to describe the spatial disposition of these primitives. The latter methods are referred to as structural and provide symbolic description of the texture. However, structural methods are not robust for noisy data. Therefore, we focus only on methods that extract descriptors from primitives.

2.3.1 Texture Unit

Considering that a texture can be seen as a set of small essential units able to characterize local information, He and Wang^{15,16} proposed the concept of texture unit, where measures computed from all units present in the texture can reveal its global aspects.

Given a 3×3 neighborhood composed by elements $V = \{v_0, v_1, \dots, v_8\}$ where v_0 represents the intensity at the central pixel and the remaining v_i represents the

intensity in its neighbors, texture unit is defined as the set $TU = \{e_1, e_2, \dots, e_8\}$, where each e_i is defined as

$$e_i = \begin{cases} 0, & \text{if } v_i < v_0 \\ 1, & \text{if } v_i = v_0 \\ 2, & \text{if } v_i > v_0 \end{cases} \quad (i = 1, 2, \dots, 8) \quad (23)$$

A signature, called texture spectrum, is obtained from the texture units. Each texture unit is associated with an index, obtained by Equation 24. There are 6561 ($= 3^8$) possible indices. The texture spectrum is a histogram with 6561 bins and the number of entries in the i -th bin, denoted by $S(i)$, is the number of texture units presenting the i -th index.

$$N_{TU} = \sum_{i=1}^8 3^{i-1} e_i \quad (24)$$

Equation 24 defines how the index of the texture unit is computed, however, it does not specify an ordering for pixels v_i , for $i > 0$. To solve that, we consider that pixels are ordered in a clockwise order starting between a and h , as shown in Figure 2. Therefore, there are eight possible orders to compute N_{TU} .

a	b	c
h		d
g	f	e

Figure 2. Neighborhood considered to compute texture unit.

He and Wang^{15,16} proposed a set of descriptors to be extracted from the texture

spectrum. The black-white symmetry (BWS) is defined as

$$\text{BWS} = 100 \left[1 - \frac{\sum_{i=0}^{3279} |S(i) - S(3281 + i)|}{\sum_{i=0}^{6560} S(i)} \right] \quad (25)$$

The descriptor called geometric symmetry (GS) computes information regarding regularity in the texture. It measures the symmetry between frequency of regions a and e , b and f , c and g , d and h , defined in Figure 2. GS is defined by Equation 26, where $S_j(i)$ contains the frequency of units with i -the index under ordering j , and j denotes one of the possible orderings used to compute the texture unit.

$$\text{GS} = 100 \left[1 - \frac{\frac{1}{4} \sum_{j=1}^4 \sum_{i=0}^{6560} |S_j(i) - S_{j+4}(i)|}{2 \sum_{i=0}^{6560} S_j(i)} \right] \quad (26)$$

Another descriptor, called degree of direction (DD), measures the degree of linearity of the primitives composing a texture. DD is defined as

$$\text{DD} = 100 \left[1 - \frac{\frac{1}{6} \sum_{m=1}^3 \sum_{n=m+1}^4 \sum_{i=0}^{6560} |S_m(i) - S_n(i)|}{2 \sum_{i=0}^{6560} S_m(i)} \right] \quad (27)$$

2.3.2 Texture Feature Coding Method

Proposed by Horng et al.,¹⁷ the texture feature coding method (TFCM) is based on the connectivity of neighbors. First-order and second-order connectivities are considered, as defined in Figure 3.

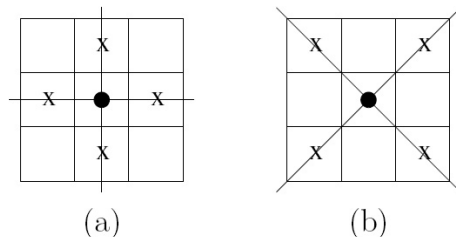


Figure 3. Connectivity defined by Horng et al.¹⁷ (a) first-order; (b) second-order.

TFCM results in pairs of integers (α, β) , where α and β compute intensity changes in the first and second-order connectivities, respectively. The changes are computed by sweeping three adjacent pixels, according to a given connectivity. Let (a, b, c) be an ordering for a sweep and (g_a, g_b, g_c) be the intensity of these pixels. If two variations between adjacent pairs is considered, there are four possible types of sweep, as defined in Equation 28, where Δ is a threshold.

$$\begin{aligned}
 & \text{(i) if } (|g_a - g_b| \leq \Delta) \text{ and } (|g_b - g_c| \leq \Delta) \\
 & \text{(ii) if } [(|g_a - g_b| \leq \Delta) \text{ and } (|g_b - g_c| \geq \Delta)] \text{ or} \\
 & \quad [(|g_a - g_b| \geq \Delta) \text{ and } (|g_b - g_c| \leq \Delta)] \\
 & \text{(iii) if } [(g_a - g_b > \Delta) \text{ and } (g_b - g_c > \Delta)] \text{ or} \\
 & \quad [(g_b - g_a > \Delta) \text{ and } (g_c - g_b > \Delta)] \\
 & \text{(iv) if } [(g_a - g_b > \Delta) \text{ and } (g_c - g_b > \Delta)] \text{ or}
 \end{aligned} \tag{28}$$

$$[(g_b - g_a > \Delta) \text{ and } (g_b - g_c > \Delta)]$$

Since each order of connectivity presents two sweeps, Figure 4 can be used to summarize each sweep according to the order of connectivity. Given these possible values for α and β , the texture feature number is defined as $TFN(x, y) = \alpha(x, y)\beta(x, y)$, resulting in 41 different values.

	(i)	(ii)	(iii)	(iv)
(i)	1	2	3	4
(ii)	2	5	6	7
(iii)	3	6	8	9
(iv)	4	7	9	10

Figure 4. Values used to compute α and β . Rows indicate the type of the first sweep and columns the type of the second sweep.

Once the TFN is computed for each pixel in the texture, a histogram and a co-occurrence matrix (Section 2.1.2) are estimated. Both are used to compute the descriptors that compose the feature vector. The histogram is defined in Equation 29, where $N(i)$ denotes the frequency of the i -th TFN and n the number of pixels in the texture. The co-occurrence matrix is computed by Equation 30, where $N_{d,\theta}(i, j)$ denotes the number of transitions between TFN with value i to j given a distance d and orientation θ .

$$p_{\text{TFCM}}(i) = \frac{N(i)}{n}, \quad n \in \{0, 1, \dots, 41\} \quad (29)$$

$$p_{\text{TFCM}}(i, j|d, \theta) = \frac{N_{d,\theta}(i, j)}{N_t}, \quad i, j \in \{0, 1, \dots, 41\} \quad (30)$$

Based on the histogram, measures of coarseness (Co), homogeneity (Ho), mean

convergence (MC) and variance (Va) are computed. These measures are defined as

$$\text{Co} = \sum_x \sum_y p_{\text{TFCM}}(41) \quad (31)$$

$$\text{Ho} = \sum_x \sum_y p_{\text{TFCM}}(0) \quad (32)$$

$$\text{MC} = \sum_{n=0}^{41} \frac{|n p_{\text{TFCM}}(n) - \mu|}{\sigma} \quad (33)$$

$$\text{Va} = \sum_{n=0}^{41} (n - \mu)^2 p_{\text{TFCM}}(n) \quad (34)$$

Using co-occurrence matrix measures of code entropy (CE) and code similarity (CS) are computed according to Equations 35 and 36.

$$\text{CE} = - \sum_{i=0}^{41} \sum_{j=0}^{41} p_{\text{TFCM}}(i, j|d, \theta) \log p_{\text{TFCM}}(i, j|d, \theta) \quad (35)$$

$$\text{CS} = \sum_{i=0}^{41} p_{\text{TFCM}}^2(i, i|d, \theta) \quad (36)$$

2.3.3 Local Binary Patterns

Local Binary Patterns (LBP) characterizes the spatial structure of a texture and presents the characteristics of being invariant to monotonic transformations of the gray-levels.¹⁸ On its standard version, a pixel c with intensity $g(c)$ is labeled as defined by Equation 37, where pixels p belong to a 3×3 neighborhood with gray

levels g_p ($p = 0, 1, 2, \dots, 7$).

$$S(g_p - g_c) = \begin{cases} 1, & g_p \geq g_c \\ 0, & g_p < g_c \end{cases} \quad (37)$$

Then, the LBP pattern of the pixel neighborhood is computed by summing the corresponding thresholded values $S(g_p - g_c)$ weighted by a binomial factor of 2^k as

$$\text{LBP} = \sum_{k=0}^7 S(g_p - g_c) 2^k \quad (38)$$

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).¹⁹ Different from the LBP, the ILBP uses the average of the 3×3 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 39.

$$\text{LBP} = \sum_{k=0}^7 S(g_p - m) 2^k + S(g_c - m) 2^8 \quad (39)$$

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

$$S(x) = \begin{cases} 1, & g_p > 0 \\ 0, & g_p \leq 0 \end{cases} \quad (40)$$

Similarly to LPB, once the labels have been computed for every pixel, an his-

togram is computed, in this case a 511-bin histogram, which will be used as feature vector.

2.3.4 Coordinated Clusters Representation

The coordinated clusters representation (CCR), proposed by Kurmyshev and Cervantes,²⁰ is a descriptor for binary texture, where the image is characterized by a histogram of occurrence of the possible binary patterns. This descriptor was later extended to grayscale texture images.²¹

A matrix of binary image intensities is denoted as $S^\alpha = \{s^{\alpha l, m}\}$, where $l = 1, 2, \dots, L$, $m = 1, 2, \dots, M$, $N = L \times M$ and $\alpha = 1, 2, \dots, 2^N$ is the index of the image. The binary image S^α is scanned with one-pixel steps by a rectangular window $W = I \times J$ pixels. Since pixels correspond to binary units, the number of all possible states of the window is 2^W .

The pixel configuration found by the window is coded as the binary number corresponding to this configuration. The coordinated cluster representation for a given binary image S^α consists of a histogram $H_{(I,J)}^\alpha(b)$ of occurrence of the pixel patterns detected by the scanning window. When the histogram $H_{(I,J)}^\alpha(b)$ is normalized, it can be interpreted as a distribution function of occurrences given by

$$F_{(I,J)}^\alpha(b) = \frac{H_{(I,J)}^\alpha(b)}{A} \quad (41)$$

2.3.5 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,²²⁻²⁴ where an operator consists in analyzing the amount of image detail removed by applying morphological openings γ_λ of increasing size λ .

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 λ , producing a granulometric curve. The normalized version of the operator for an image f can be written as

$$G(f) = \frac{\text{Vol}(\gamma_\lambda(f))}{\text{Vol}(f)} \quad (42)$$

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

2.4 Approach Based on Parametric Models

In this approach, a texture is considered as a sample from a stochastic process defined by a set of parameters, which are used to summarize the texture. Analysis and synthesis of texture can be performed with the use of these parameters. This section describes parametric models based on Markov random fields^{25,26} and simultaneous autoregressive models.^{27,28}

2.4.1 Markov Random Fields

Texture analysis based on Markov random fields (MRF) uses the set of parameters estimated from the probability distribution as descriptors. The model of local descriptors proposed by Ising²⁹ was used by Cross and Jain³⁰ to model textures considering neighborhoods of first, second, third, and fourth orders. The Ising model,

originally proposed to empirically explain observed facts about ferromagnetic materials, is an important and simple representation in MRF, where a lattice of points can be used to model the interaction of neighboring points.

Local descriptors are defined by Equation 43, where parameters q depend on the neighborhood order, which is defined according to the scheme shown in Figure 5.

$$P(X_s = x_s | \partial_s) = \frac{\exp(-qx)}{1 + \exp(-q)} \quad (43)$$

where $S = \{1, 2, \dots, n\}$, X_s is a random variable, and ∂_s is a collection of neighbors in S .

	o_1	m	q_1	
o_2	v	u	z	q_2
l	t	s	t'	l'
q'_1	z'	u'	v'	o'_1
	q'_2	m'	o'_2	

Figure 5. Representation of the neighborhood of a central pixel with intensity x_s .

The first order neighborhood model uses q defined as

$$q_1 = a + b_{1,1}(x_t + x_{t'}) + b_{1,2}(x_u + x_{u'}) \quad (44)$$

the second order neighborhood

$$q_2 = q_1 + b_{2,1}(x_v + x_{v'}) + b_{2,2}(x_z + x_{z'}) \quad (45)$$

the third order neighborhood

$$q_3 = q_2 + b_{3,1}(x_m + x_{m'}) + b_{3,2}(x_l + x_{l'}) \quad (46)$$

and, finally, for the fourth order

$$q_4 = q_3 + b_{4,1}(x_{o_1} + x_{o'_1} + x_{o_2} + x_{o'_2}) + b_{4,2}(x_{q_1} + x_{q'_1} + x_{q_2} + x_{q'_2}) \quad (47)$$

Coefficients a and $b_{i,j}$ are used as texture descriptors. Their estimation is performed using the coding method proposed in.²⁵

2.4.2 Simultaneous Autoregressive Models

As an instance of the MRF models, the simultaneous autoregressive models (SAR) have been successfully applied to texture classification by considering the spatial interactions among neighboring pixels to represent textures.²⁷ Therefore, similarly to MRF, a SAR model for a pixel is defined as a function of its neighbors, as shown in Equation 48, where $f(x, y)$ is the image intensity at location (x, y) , N defines its neighborhood, $\theta(k, l)$ are the model parameters and $E(m, n)$ is an error term associated to the pixel.

$$f(x, y) = \sum_{(k,l) \in N} \theta(k, l) f(m - k, n - l) + E(m, n) \quad (48)$$

Given a texture, parameters $\theta(k, l)$ are estimated considering small neighborhoods around each pixel and, in this work the parameter estimation is performed using the

Least Square Error (LSE) technique. Then, the concatenation of these parameters composes the feature vector used to describe the texture.

2.5 Summary of Texture Descriptors

Table 1 presents a summary of the texture descriptors described in the previous sections. Main parameters and number of dimensions are shown for each texture descriptor categorized into its respective approach.

Approach	Descriptor	Parameters	Reference (Section)	Dimension
Statistical: texture properties are represented in indirect and probabilistic manners.	First order Statistics	-	2.1.1	6
	GLCM	orientations, distance between two pixels and quantization levels	2.1.2	48
	GLRLM	orientations	2.1.3	10
	Autocorrelation	-	2.1.4	81
	Fourier Spectrum	-	2.2.1	2
Signal Processing: image transforms are applied to extract feature descriptors.	Wavelets	wavelet basis	2.2.2	6
	Gabor Filters	number of scales and orientations	2.2.3	80
	Texture Unit	-	2.3.1	3
Geometrical: descriptors are extracted from texture primitives (textels) initially identified.	TFCM	Δ , GLCM displacement	2.3.2	6
	LBP	-	2.3.3	256
	ILBP	-	2.3.3	512
	CCR	window size	2.3.4	16, 256, 65536
	Granulometry	structuring element size, step to increase the structuring element	2.3.5	24-104
Parametric Models: texture is considered as a sample from a stochastic process defined by a set of parameters used to describe the texture.	MRF	order of the neighborhood	2.4.1	3-9
	SAR	size of the neighborhood	2.4.2	9

Table 1. Summary of feature descriptors classified according to their approaches.

3. EXPERIMENTAL RESULTS

This section describes the experiments conducted to evaluate the feature descriptors applied to texture classification. We consider four texture data sets: UIUC,³¹ UMD,³² Outex³³ and VisTex.³⁴ All data sets were used to estimate and evaluate the parameters employed to assess the effectiveness of the feature descriptors for texture classification.

In Section 3.1, we describe the setup used for each method. Section 3.2 presents a brief explanation of each data set used in our experiments. Then, the feature extraction methods are compared and discussed in Section 3.3.

3.1 Experimental Setup

Details of implementation for the feature extraction methods as well as the parameters used by each are given as follows.

Markov Random Fields. We consider first, second, third, and fourth neighborhood orders for the generalized Ising model used for the MRF, having 3, 5, 7, and 9 parameters, respectively.

Autocorrelation. The implementation of the autocorrelation function possesses two parameters to be estimated, p and q . These parameters are directly related to the number of variables in the feature vector. Their optimum values are experimentally estimated. According to the experiments, parameters $p = q = 9$ achieved the best results and will be considered in the remaining experiments.

Gray Level Co-occurrence Matrix. A subset of the 14 descriptors described in Section 2.1.2 is considered for the co-occurrence matrices: 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. The optimum value for parameter d is experimentally estimated to be 1 and four orientations are considered and concatenated, resulting a feature vector with 48 dimensions.

Gray Level Run Length Matrices. Regarding the gray level run length matrices, we consider all descriptors presented in Section 2.1.3 for orientation of 0° and 90° , therefore, resulting in a feature vector composed of 10 variables.

Coordinated Clusters Representation. The CCR method has the window size as free parameter.

Granulometry. The parameters considered for the granulometry are the kernel size, using morphological opening and closing operations, and the step parameter, which indicates the increment in size at a time.

Texture Feature Coding Method. Experiments were performed to estimate the best value for the parameters Δ and the displacement of the co-occurrence matrix, described in Section 2.3.2.

Wavelets. We used the Daubechies wavelet basis with two levels of decomposition in the wavelet-based method. The energy coefficients are obtained from the sub-images with high frequency. Therefore, the feature vector has 6 variables. In the experiments, we considered two wavelet bases. The inputs are necessarily square images, otherwise they will be cropped to the largest possible square.

Fourier Spectrum. As a restriction of this method, the inputs must be power of 2 grayscale images, otherwise they will be cropped.

Simultaneous Autoregressive Models. Parameters k_1 and k_2 are used to indicate the size of the neighborhood. The experiments were conducted using $k_1 = k_2$, varying k_1 between 6 to 10. The feature vector has 9 dimensions.

3.2 Data Sets

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

3.2.1 UMD Data Set

The UMD high-resolution data set³² contains images of 1280×960 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 6.

3.2.2 UIUC Data Set

The UIUC data set³¹ is a texture data set composed of 1000 images of 640×480 pixels, distributed in 25 classes (variations of wood, gravel, fur, carpet, brick, among others) with 40 samples each. Figure 7 shows examples of images for each texture.

3.2.3 OuTex Data Set

OuTex³⁵ is a framework for evaluation of texture classification and segmentation. It contains several images and protocols for texture classification. The images are

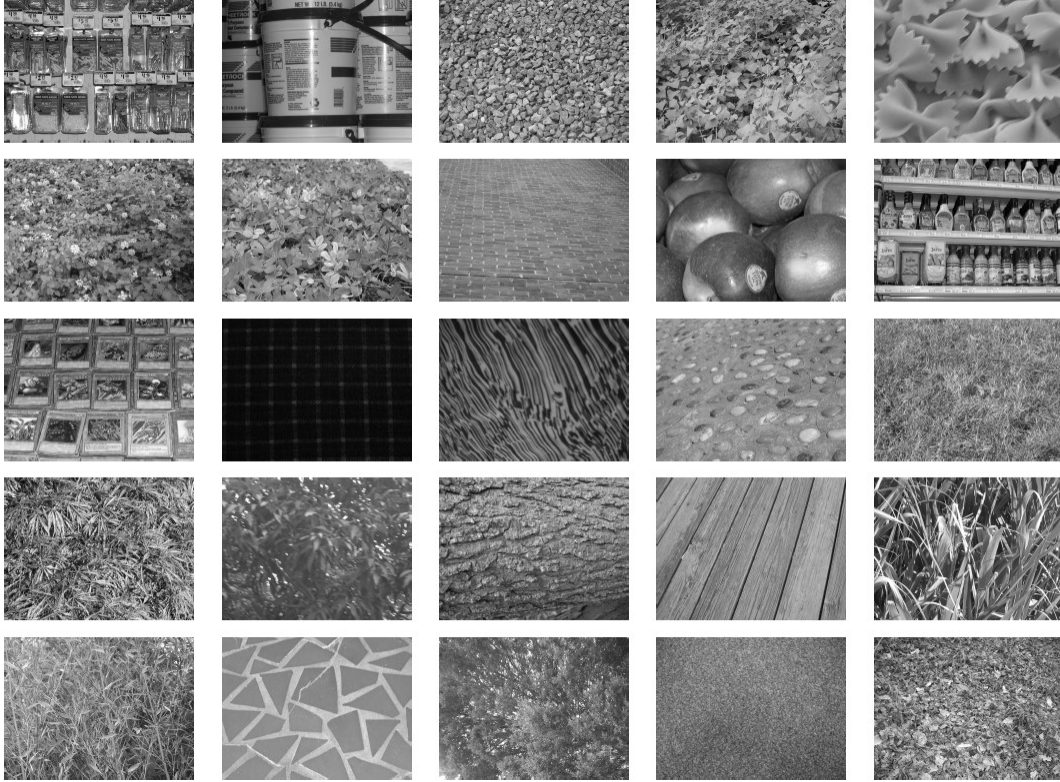


Figure 6. Examples of 25 texture samples extracted from UMD data set.³²

acquired under several illuminations and different angles, over a number of different textures.

Several test suites for texture classification have been proposed. Out of 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, and a hundred of different arrangements for training and testing. Figure 8 shows examples of texture images.

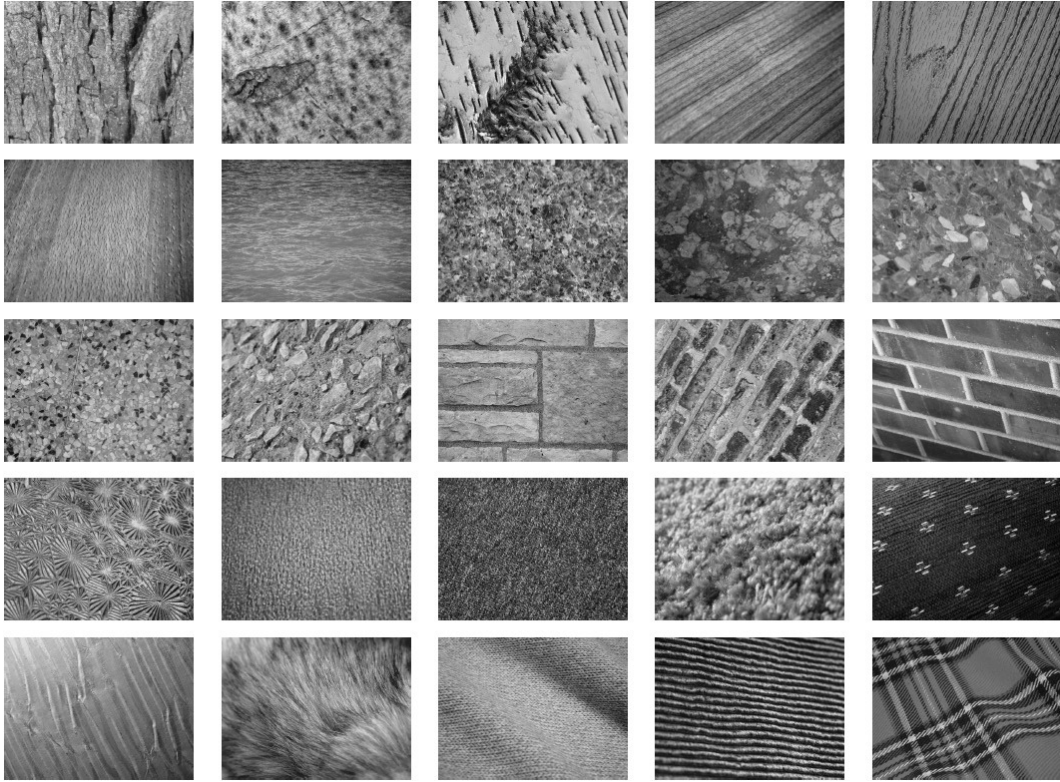


Figure 7. Examples of 25 texture samples extracted from UIUC data set.³¹

3.2.4 VisTex Data Set

VisTex³⁴ 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.³⁶ Such images are available on the Outex site³⁵ as test suite Contrib_TC_00006. For each texture class, half of the samples were used in the training set and the other half were used as testing data. Figure 9 shows examples of texture images.

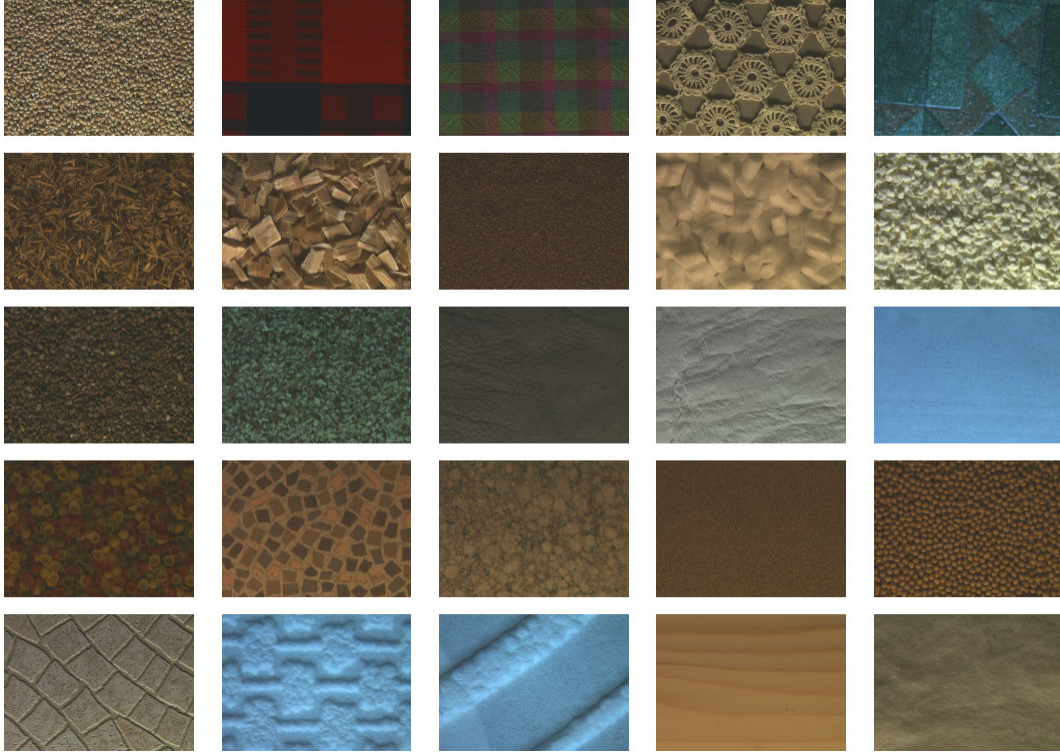


Figure 8. Examples of texture samples extracted from OuTex data set.³⁵

3.3 Results and Comparisons

To perform texture classification, each data set was partitioned into two sets: training and test. One hundred random splits were employed for UMD and UIUC datasets, considering five, ten, fifteen and twenty training samples, whereas the remaining samples were used as test. For OuTex and VisTex data sets, the test suites TC_00005³⁵ and Contrib_TC_00006³⁴ were considered in our experiments, respectively.

The reported results correspond to the average of all the considered splits. Nearest neighbor (1-NN) classifier is applied after all variables are normalized to present zero mean and unit variance. It is worth pointing out that there was no contamination between training and test patterns in the experiments. The parameter estimation for

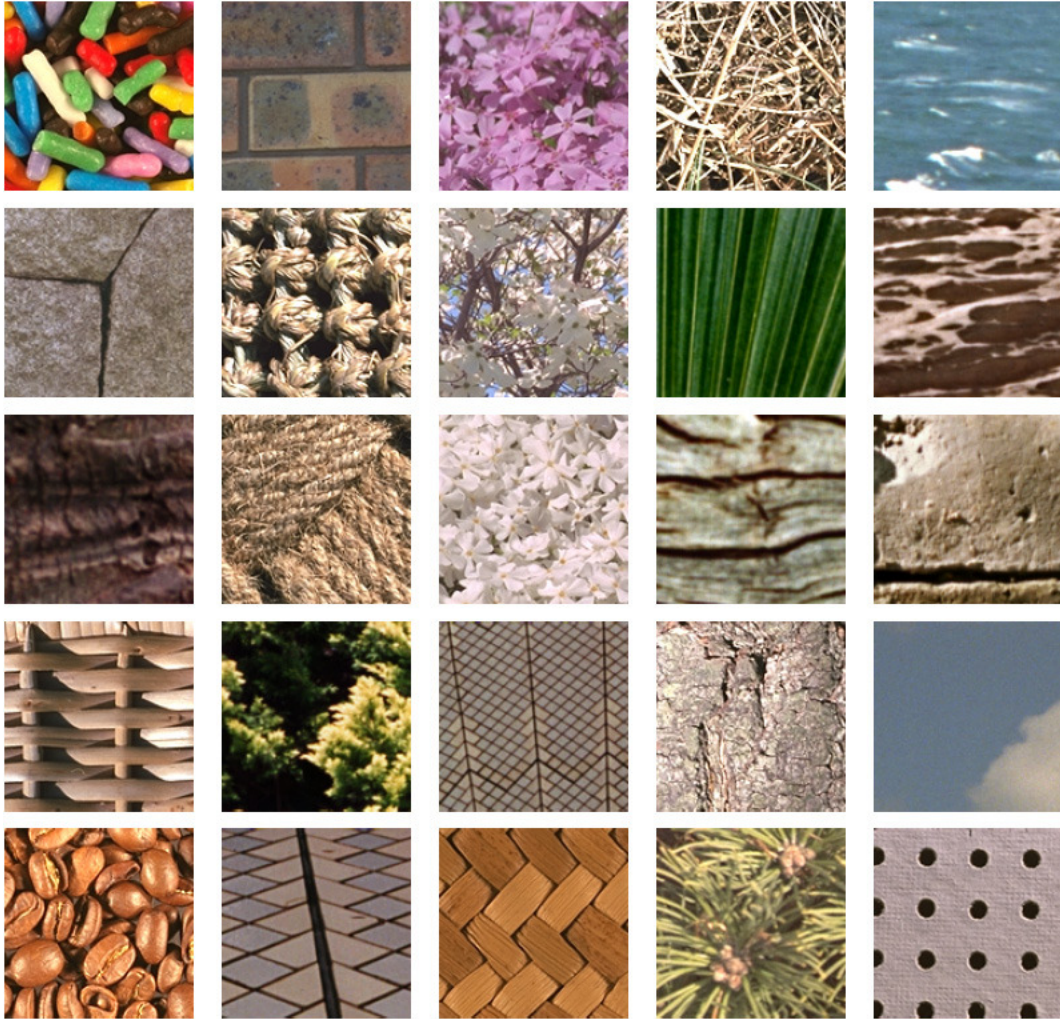


Figure 9. Examples of texture samples extracted from Vistex data set.³⁴

the normalization (mean and variance) is performed by using only patterns belonging to the training set.

Dimensionality is reduced by using Principal Component Analysis (PCA). Since 1-NN is used as classifier and it relies on the distance between feature descriptors extracted from the patterns, the direct use of the classifier without reducing the

dimensionality of the data with PCA would suffer from the curse of dimensionality, in which the feature space becomes sparser as the dimensionality increases.

Four distance functions are considered to measure the similarity between pairs of samples: Euclidean, city block, cosine, and correlation. Besides the results achieved using single feature extraction methods, results considering feature combination are also presented. The combination of the features is constructed by concatenating all features together.

Tables 2 to 5 show the classification rates for each data set considering different feature parameters. The best results achieved for each feature extraction method are displayed in Figure 10. This figure summarizes the results for all data sets showing in the x -axis normalized results (the closer to 100, the better the results achieved by a given feature extraction method). Methods ILBP, LBP, Gabor, GLCM and granulometry are the methods that achieved the best results throughout all experiments performed. Finally, it is important to point out that the feature combination achieved the highest texture classification rates for all data sets.

4. CONCLUSIONS

Texture information plays an important role in several domains, such as image segmentation, image classification, content-based image retrieval, scene recognition, video analysis, among others. Although several texture descriptors have been proposed in the literature, the formal definition of texture in images is still a complex task.

Features	Parameters		Classification Rates (%)			
			5	10	15	20
First order statistics	-		50.272	53.582	55.328	56.256
GLCM	Θ 0°, 45°, 90°, 135°	d level 1 256	82.178	87.898	90.524	92.132
GLRLM	Θ 0° and 90°		62.068	69.420	72.603	74.698
Autocorrelation	-		55.772	64.317	69.553	72.676
Fourier spectrum	-		18.800	18.689	18.692	18.492
Wavelets	basis					
	db4		53.088	61.734	66.105	69.526
Gabor filters	db8		55.080	62.633	66.464	69.146
	scale	Θ				
Texture unit	5	8	75.695	82.329	84.857	88.098
TFCM	-		34.859	36.777	38.217	39.168
TFCM	Δ	d				
	0	1	55.420	61.885	64.995	67.516
	0	2	56.600	63.110	66.467	69.174
	0	3	56.648	63.256	66.563	69.370
	0	4	56.502	62.750	66.300	68.792
	0	5	55.818	62.301	65.545	68.026
	1	1	59.065	66.406	69.968	72.472
	1	2	59.006	66.965	70.808	73.762
	1	3	58.524	66.562	70.227	73.356
	1	4	58.126	66.166	70.134	73.346
1	5	57.658	65.576	69.438	72.482	
LBP	-		72.522	81.501	85.468	88.188
ILBP	-		73.356	81.366	85.081	87.432
CCR	window					
	3		68.996	77.509	81.756	84.276
Granulometry	4		68.794	77.494	81.800	84.334
	kernel size	step				
	5	1	69.996	79.114	83.956	86.372
	5	2	70.609	79.756	84.515	87.124
	10	1	73.461	82.340	87.035	89.504
	10	2	73.569	82.345	86.982	89.490
	15	1	73.601	82.468	87.201	89.840
	15	2	73.893	82.789	87.561	90.140
	20	2	74.386	83.224	87.708	90.246
	25	2	75.435	83.861	88.076	90.698
MRF	order					
	0		30.845	33.246	34.809	35.894
	1		48.506	55.076	60.632	63.366
	2		57.272	65.117	70.748	74.184
SAR	3		61.336	69.560	74.347	77.738
	size					
	6		65.298	73.606	77.660	80.308
	7		64.082	72.486	76.825	79.342
	8		62.225	70.805	75.508	77.902
Feature combination	9		60.219	69.188	74.096	76.780
	10		58.030	67.402	72.296	75.168
Feature combination	-		85.01	89.448	85.006	92.08

Table 2. Results of the experiments on the UMD data set as function of the number of training samples.

Features	Parameters		Classification Rates (%)			
			5	10	15	20
First order statistics	-		40.536	44.465	44.108	48.548
GLCM	Θ 0°, 45°, 90°, 135°	d level 1 256	53.817	62.269	66.761	69.642
GLRLM	Θ 0° and 90°		26.180	28.850	30.777	31.804
Autocorrelation	-		29.893	35.220	38.752	40.988
Fourier spectrum	-		13.750	13.445	13.475	13.400
Wavelets	basis					
	db4		26.216	30.341	32.979	34.392
Gabor filters	db8		25.229	28.833	31.216	36.172
	scale	Θ				
Texture unit	5	8	50.961	59.352	65.110	69.861
Texture unit	-		22.948	24.780	24.899	25.328
TFCM	Δ	d				
	0	1	24.461	27.244	28.484	29.060
	0	2	24.340	27.032	28.123	28.552
	0	3	24.667	27.521	28.827	29.296
	0	4	25.056	27.872	29.505	29.956
	0	5	25.012	27.797	29.377	29.980
	1	1	27.342	31.424	33.038	34.446
	1	2	27.157	31.380	33.168	34.462
	1	3	27.248	31.549	33.372	34.900
	1	4	27.429	31.552	33.558	34.948
	1	5	27.419	31.656	33.673	35.056
LBP	-		41.712	44.681	49.272	53.489
ILBP	-		44.808	51.737	54.765	61.327
CCR	window					
	3		34.427	40.752	44.361	46.748
	4		34.420	40.774	44.388	46.802
Granulometry	kernel size	step				
	5	1	40.934	50.266	55.366	59.238
	5	2	41.737	50.701	55.651	59.210
	10	1	44.481	54.433	59.750	63.980
	10	2	44.556	54.321	59.691	63.746
	15	1	45.594	56.154	61.604	65.610
	15	2	46.196	56.933	62.395	66.306
	20	2	47.024	57.821	63.276	67.252
	25	2	48.353	58.924	64.592	68.674
MRF	order					
	0		19.356	20.448	20.780	21.274
	1		27.438	30.637	32.484	34.284
	2		30.004	35.129	38.057	40.204
	3		28.581	33.844	36.851	39.032
SAR	size					
	6		28.217	33.912	37.190	39.842
	7		26.138	31.278	34.054	36.672
	8		23.667	28.228	30.715	33.254
	9		21.825	25.605	27.838	30.184
	10		20.158	23.432	25.054	27.072
Feature combination	-		54.807	63.921	68.683	71.842

Table 3. Results of the experiments on the UIUC data set as function of the number of training samples.

Features	Parameters	Classification Rates (%)
First order statistics	-	44.675
GLCM	Θ d level 0°, 45°, 90°, 135° 1 256	94.444
GLRLM	Θ 0° and 90°	63.888
Autocorrelation	-	76.620
Fourier spectrum	-	14.814
Wavelets	basis db4 db8	78.703 80.092
Gabor filters	scale Θ 5 8	92.592
Texture unit	-	32.638
TFCM	Δ d 0 1 0 2 0 3 0 4 0 5 1 1 1 2 1 3 1 4 1 5	67.129 64.583 62.037 63.194 62.735 72.685 71.759 71.759 68.750 71.527
LBP	-	96.527
ILBP	-	97.916
CCR	window 3 4	80.787 80.787
Granulometry	kernel size step 5 1 5 2 10 1 10 2 15 1 15 2 20 2 25 2	93.518 93.750 94.444 93.518 91.203 91.666 91.435 88.657
MRF	order 0 1 2 3	30.324 61.574 74.074 83.564
SAR	size 6 7 8 9 10	80.787 71.990 66.435 61.111 58.564
Feature combination	-	99.768

Table 4. Results of the experiments on the VisTex data set.

Features	Parameters	Classification Rates (%)
First order statistics	-	51.205
GLCM	Θ d level 0°, 45°, 90°, 135° 1 256	83.147
GLRLM	Θ 0° and 90°	59.703
Autocorrelation	-	40.265
Fourier spectrum	-	10.441
Wavelets	basis db4 db8	52.513 49.181
Gabor filters	scale Θ 5 8	84.912
Texture unit	-	25.995
TFCM	Δ d 0 1 0 2 0 3 0 4 0 5 1 1 1 2 1 3 1 4 1 5	44.789 42.053 42.121 39.188 37.862 43.152 40.591 39.471 37.242 37.135
LBP	-	89.044
ILBP	-	82.091
CCR	window 3 4	69.353 69.523
Granulometry	kernel size step 5 1 5 2 10 1 10 2 15 1 15 2 20 2 25 2	89.266 87.428 87.376 87.007 86.723 86.576 85.139 83.713
MRF	order 0 1 2 3	20.692 30.857 41.658 42.629
SAR	size 6 7 8 9 10	49.555 45.338 41.378 35.777 31.605
Feature combination	-	93.922

Table 5. Results of the experiments on the OuTex data set.

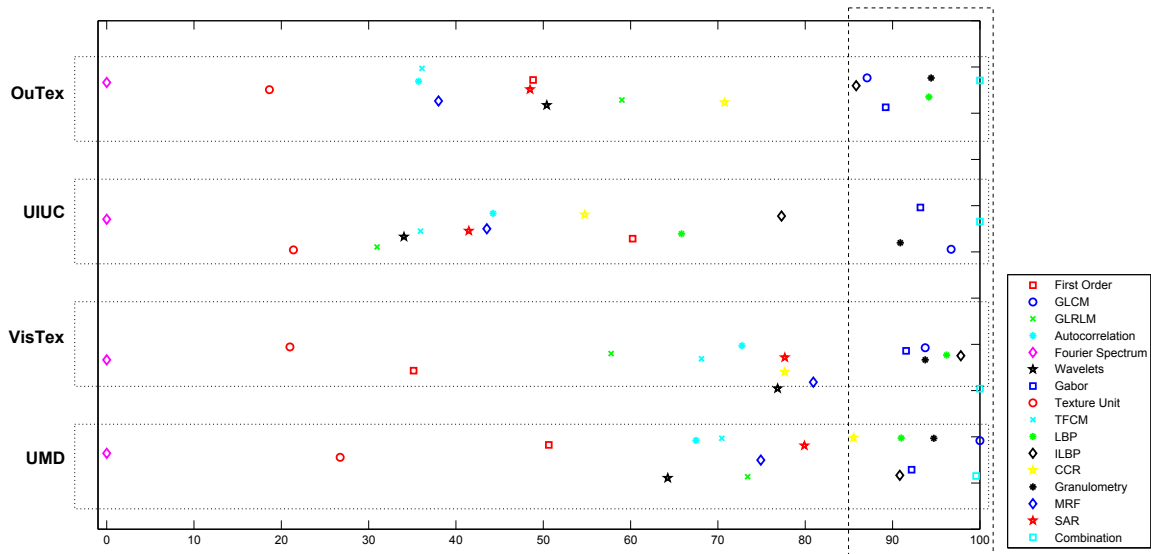


Figure 10. Comparison of results for all considered features on each data set. Each horizontal dotted box displays results for one data set. The x -axis shows normalized results (between 0 and 100) according to the minimum and maximum classification rates achieved for each data set. The vertical slashed box on the right-hand side selects the feature extraction methods that achieved the best results throughout all data sets.

This work presented and compared a set of texture descriptors to assess their suitability for texture classification. Statistical, signal processing, geometrical, and parametric model approaches are used to extract texture features from the images.

Experiments demonstrated that feature descriptors such as ILBP, LBP, Gabor filters, GLCM and granulometry present consistently high classification rates in different data sets. Furthermore, the combination of features, formed by the concatenation of them together, produced superior results in terms of classification rate when compared to the use of the features individually.

ACKNOWLEDGMENTS

The authors are grateful to FAPESP, CNPq and CAPES for the financial support. This research was partially supported by FAPESP Grant 2010/10618-3.

REFERENCES

- [1] Schwartz, W. and Pedrini, H., “Textured Image Segmentation Based on Spatial Dependence Using a Markov Random Field Model,” in [*2006 IEEE International Conference on Image Processing*], 2449–2452 (Oct. 2006).
- [2] Haralick, R. M., “Statistical and Structural Approaches to Texture,” *Proceedings of the IEEE* **67**, 786–804 (May 1979).
- [3] Reed, T. and Dubuf, J. M. H., “A Review of Recent Texture Segmentation and Feature Extraction Techniques,” *CVGIP: Image Understanding* **57**, 359–372 (May 1993).
- [4] Xu, Y., Huang, S. B., and Ji, H., “Integrating Local Feature and Global Statistics for Texture Analysis,” in [*16th IEEE International Conference on Image Processing*], 1377–1380 (Nov. 2009).
- [5] Tamura, H., S, M., and Yamawaki, Y., “Textural Features Corresponding to Visual Preception,” *IEEE Transactions on Systems, Man, and Cybernetics* **8**, 237–247 (1978).
- [6] Lasmar, N.-E., Stitou, Y., and Berthoumieu, Y., “Multiscale Skewed Heavy Tailed Model for Texture Analysis,” in [*16th IEEE International Conference on Image Processing*], 2281–2284 (Nov. 2009).
- [7] Tao, Z., Wenxue, H., and Jinjia, W., “A Novel Texture Analysis Method Based on Graph Spectral Theory,” in [*Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*], 467–470 (Sept. 2009).

- [8] Tuceryan, M. and Jain, A. K., “Texture Analysis,” in [*The Handbook of Pattern Recognition and Computer Vision*], Chen, C. H., Pau, L. F., and Wang, P., eds., 207–248, World Scientific Publishing Co. (Aug. 1998).
- [9] Haralick, R. M., Shanmugam, K., and Dinstein, I., “Textural Features for Image Classification,” *IEEE Transactions on Systems, Man and Cybernetics* **3**, 610–621 (Nov. 1973).
- [10] Galloway, M. M., “Texture Analysis Using Gray Level Run Lengths,” *Computer Graphics and Image Processing* **4**, 172–179 (1975).
- [11] Daubechies, I., [*Ten Lectures on Wavelets*], CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992).
- [12] Mallat, S. G., “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, 674–693 (July 1989).
- [13] Unser, M., “Texture Classification and Segmentation using Wavelet Frames,” *IEEE Transactions on Image Processing* **4**, 1549 – 1560 (Nov. 1995).
- [14] Daugman, J. G., “Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters,” *Journal of the Optical Society of America A* **2**, 1160–1169 (1985).
- [15] He, D. C. and Wang, L., “Texture Unit, Texture Spectrum, and Texture Analysis,” *IEEE Transactions on Geoscience and Remote Sensing* **28**, 509–512 (July 1990).
- [16] He, D. C. and Wang, L., “Texture Features Based on Texture Spectrum,” *Pattern Recognition* **24**, 391–399 (May 1991).

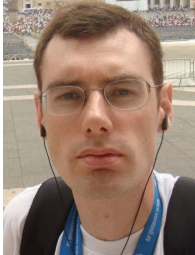
- [17] Horng, M.-H., Sunb, Y.-N., and Lin, X.-Z., “Texture Feature Coding Method for Classification of Liver Sonography,” *Computerized Medical Imaging and Graphics* **26**, 33–42 (Jan. 2002).
- [18] Ojala, T., Pietikinen, M., and Harwood, D., “A Comparative Study of Texture Measures with Classification Based on Featured Distributions,” *Pattern Recognition* **29**, 51–59 (Jan. 1996).
- [19] Jin, H., Liu, Q., Lu, H., and Tong, X., “Face Detection Using Improved LBP under Bayesian Framework,” in [*Proceedings of the Third International Conference on Image and Graphics*], 306–309 (2004).
- [20] Kurmyshev, E. and Cervantes, M., “A Quasi-Statistical Approach to Digital Image Representation,” *Revista Mexicana de Física* **42**(1), 104–116 (1996).
- [21] Sanchez-Yanez, R., Kurmyshev, E., and Cuevas, F., “A Framework for Texture Classification using the Coordinated Clusters Representation,” *Pattern Recognition Letters* **24**(1–3), 21–31 (2003).
- [22] Hanbury, A., Kandaswamy, U., and Adjeroh, D. A., “Illumination-Invariant Morphological Texture Classification,” in [*Mathematical Morphology: 40 Years On*], et al., C. R., ed., 377–386, Springer, The Netherlands (2005).
- [23] Serra, J., [*Image Analysis and Mathematical Morphology*], Academic Press, New York, NY, USA (1982).
- [24] Heijmans, H. J. A. M., [*Morphological Image Operators*], Academic Press, New York, NY, USA (1994).
- [25] Besag, J. E., “Spatial Interaction and the Statistical Analysis of Lattice Systems,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **36**, 192–236 (May 1974).

- [26] Geman, S. and Geman, D., “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741 (Nov. 1984).
- [27] Mao, J. and Jain, A. K., “Texture Classification and Segmentation using Multiresolution Simultaneous Autoregressive Models,” *Pattern Recognition* **25**, 173–188 (1992).
- [28] Randen, T. and Husøy, J. H., “Filtering for Texture Classification: A Comparative Study,” *IEEE Transaction on Pattern Analysis Machine Intelligence* **21**, 291–310 (1999).
- [29] Ising, E., “Beitrag zur Theorie des Ferromagnetismus,” *Zeitschrift für Physik A Hadrons and Nuclei* **31**(1), 253–258 (1925).
- [30] Cross, G. R. and Jain, A. K., “Markov Random Field Texture Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **5**, 25–39 (Jan. 1983).
- [31] Lazebnik, S., Schmid, C., and Ponce, J., “A Sparse Texture Representation Using Local Affine Regions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, 1265–1278 (2005).
- [32] UMD, “Dataset,” (2011). http://www.cfar.umd.edu/~fer/High-resolution-data-base/hr_database.htm.
- [33] Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J., and Huovinen, S., “Outex - New Framework for Empirical Evaluation of Texture Analysis Algorithms,” in [*Proceedings of the 16th International Conference on Pattern Recognition*], **1** (2002).
- [34] VisTex, “Dataset,” (2011). <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.
- [35] OuTex, “Dataset,” (2011). http://www.outex.oulu.fi/index.php?page=classification#Outex_TC_00005.

- [36] Arvis, V., Debain, C., Berducat, M., and Benassi, A., “Generalization of the Cooccurrence Matrix for Colour Images: Application to Colour Texture Classification,” *Image Analysis & Stereology* **23**, 63–72 (2004).

Author Biographies

William Robson Schwartz received his Ph.D. degree in Computer Science from University of Maryland, College Park, USA. He received his B.Sc. and M.Sc. degrees in Computer Science from the Federal University of Parana, Curitiba, Brazil. He is currently a professor in the Department of Computer Science at the Federal University of Minas Gerais, Brazil. His research interests include computer vision, biometrics and image processing.



Fernando Roberti de Siqueira received his B.Sc. degree in Computer Science from the University of Campinas, Campinas, Brazil. He is currently a M.Sc. student in the Institute of Computing at the University of Campinas, Brazil. His research interests include image processing, computer vision, data mining and machine learning.



Helio Pedrini received his Ph.D. degree in Electrical and Computer Engineering from Rensselaer Polytechnic Institute, Troy, NY, USA. He received his M.Sc. degree in Electrical Engineering and his B.Sc. in Computer Science, both from the University of Campinas, Brazil. He is currently a professor in the Institute of Computing at the University of Campinas, Brazil. His research interests include image processing, computer vision, pattern recognition, computer graphics, computational geometry.

