Pattern Recognition Letters 34 (2013) 1314–1319

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Texture analysis and classification using shortest paths in graphs

Jarbas Joaci de Mesquita Sá Junior[a,1], André Ricardo Backes[b,*,1], Paulo César Cortez[a]

[a] Departamento de Engenharia de Teleinformática – DETI, Centro de Tecnologia – UFC, Campus do Pici, S/N, Bloco 725, Caixa Postal 6007, CEP 60.455-970, Fortaleza, Brazil
[b] Faculdade de Computação, Universidade Federal de Uberlândia, Av. João Naves de Ávila, 2121, 38408-100, Uberlândia, MG, Brazil

ARTICLE INFO

Article history:
Received 23 August 2012
Available online 26 April 2013

Communicated by S. Aksoy

Keywords:
Texture analysis
Graph
Shortest path

ABSTRACT

Texture is a very important attribute in the field of computer vision. This work proposes a novel texture analysis method which is based on graph theory. Basically, we convert the pixels of an image into vertices of an undirected weighted graph and explore the shortest paths between pairs of pixels in different scales and orientations of the image. This procedure is applied to Brodatz's textures and UIUC texture dataset in order to evaluate its capacity of discriminating different kinds of textures. The best classification results using the standard parameters of the method are $98.50\%, 67.30\%$ and 88.00% of success rate (percentage of samples correctly classified) for Brodatz's textures, UIUC textures (image size of 200×200 pixels), and original UIUC textures (image size of 640×480 pixels), respectively. These results prove that the proposed approach is an efficient tool for texture analysis, once they are superior to the results achieved by traditional and novel texture descriptors presented in literature.

1. Introduction

Image analysis plays an important role in the computer vision area. It is responsible for extracting meaningful information from images. Among all characteristics present in an image, texture is one of the most important and a rich source of information, being an essential attribute in many application areas, such as object recognition, remote sensing, content-based image retrieval and so on.

Texture has been the focus of a large amount of research in Psychophysics (Julesz, 1975; Beck et al., 1987). This is due to the apparent ability of a human to distinguish textures. However, the automated description and recognition of texture patterns have proven to be quite complex. In fact, there is no formal definition in literature that is capable of fully explaining it (Ebert et al., 1994; Emerson et al., 1999). Many artificial textures are characterized by the repetition of a model (in its exact form or with small variations) over a region (Backes et al., 2009). Differently, natural patterns, such as the surface of a plant leaf, present random and persistent stochastic patterns, which give rise to a cloud like appearance (Kaplan, 1999). Even the absence of patterns can characterize a texture (e.g., a noisy region in an image).

Despite its lack of definition, many methods of texture analysis have been developed over the years, each one exploring a novel approach to extract the image's texture information. For instance, we have classical methods based on second-order statistics (such as co-occurrence matrices) (Haralick, 1979; Murino et al., 1998), spectral analysis (Fourier and Gabor filters etc.) (Casanova et al., 2009; Manjunath and Ma, 1996; Azencott et al., 1997) and wavelet transform (Sengür et al., 2007; Lu et al., 1997). More recently, some works have proposed alternative ways of exploring texture images in order to fill the gaps left by other methods, such as approaches based on trajectories produced by deterministic walkers (Backes et al., 2010), fractal dimension (Backes et al., 2009; Chen and Bi, 1999; Tricot, 1995), complex network theory (Costa et al., 2007), and simplified gravitational systems (Sá Junior and Backes, 2011, 2012).

In order to improve the accuracy of texture analysis, in this paper we propose a novel texture descriptor (called Shortest Paths in Graphs method - SPG method) that explores a texture as if it were a landscape with hills, plateaus, valleys etc. Thus, a texture can be described by statistical moments obtained from shortest paths in this landscape between different pairs of points. In fact, texture is a characteristic capable of representing the physical properties of the surface of an object, that is, texture is directly related to the object's surface.

Even though there are some papers in literature that employ graphs for segmentation/classification of images, none of them extracts information by the pioneering approach proposed in this paper. For instance, the paper Jagannathan and Miller (2002) proposes a method for segmenting textures by constructing a fully connected, weighted, undirected graph, whose weights are obtained from wavelet packet generated features. Another example is the paper Xu and Xen (2004), which uses a three-dimensional information obtained from a texture image function to construct a graph, from which four descriptors are extracted to characterize

* Corresponding author. Tel.: +55 34 3239 4393; fax: +55 34 3239 4392.
E-mail addresses: jarbas_joaci@yahoo.com.br (J.J.M. Sá Junior), backes@facom.ufu.br (A.R. Backes), cortez@lesc.ufc.br (P.C. Cortez).
[1] Tel./fax: +55 85 288 9467.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.04.013

textures. The paper Jirik et al. (2011) segments textures by applying a bank of Gabor filters to measure texture similarity and uses this measure as input of a graph cut method.

The remaining of the paper is organized as follows. Section 2 shows the considerations to compute the shortest path in a graph, as also how a texture image can be modeled as a graph. Section 3 defines how to use the shortest paths of an image graph to compose a signature capable of describing the original image. In Section 4, we describe an experiment in which our approach is compared against other texture analysis methods found in literature using two benchmark texture databases. Section 5 presents the results achieved by each method evaluated and a discussion as well. Finally, we made some remarks about this paper in Section 6.

2. Considerations on shortest paths in graphs

2.1. Graphs

In the history of mathematics, the solution of Königsberg bridges problem by the Swiss mathematician Euler in 1736 originated the Graph Theory (Euler, 1736). Graphs are versatile data structures that can represent a large number of different situations and events from many different domains (Drozdek, 2000). Intuitively, a graph is a collection of vertices (or nodes) and the connections between them. A graph $G(V,E)$ consists of a set V of vertices and a set $E \subseteq V \times V$ of edges. We denote the number of vertices and edges by $|V|$ and $|E|$, respectively.

An edge presents the form $(v_i, v_j), v_i, v_j \in V$. Graphs can be undirected or directed. An undirected graph is characterized by the lack of orientation, i.e., the edge just connects the vertices without any consideration about the starting and ending vertice, $(v_i, v_j) = (v_j, v_i)$. On the other hand, a directed graph (or digraph) is the one in which edges have orientation, i.e., the information about the starting and ending edge is considered, $(v_i, v_j) \neq (v_j, v_i)$. Furthermore, a graph can be weighted if a value $w \in W : E \rightarrow \Re$ is assigned to each edge. Such weights might represent, for example, distances between cities, latency between computers of a network, and so on.

2.2. Shortest path

Graph theory considers a path as a sequence of edges $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$ connecting a *starting vertice*, v_1, to an *ending vertice*, v_n. The path can also be represented by the set of vertices between these two vertices: $\{v_1, v_2, \ldots, v_n\}$.

Finding the shortest path between two vertices is a classical problem in graph theory. Over the years, a large number of solutions have been proposed. The simplest case is when all the weights of a graph are equal and non-zero. For such graph, a Breadth First Search (BFS) algorithm can find the shortest path. On the other hand, in graphs with negative edge weights the Bellman-Ford algorithm (Bellman, 1958) is necessary for solving the problem. In this work, we propose to model the texture as an undirected weighted graph with $w \geqslant 0$, so that we employ Dijkstra's algorithm, once this algorithm finds the shortest path between two vertices with the small computational complexity.

2.3. Dijkstra's algorithm

Conceived by the Dutch computer scientist Dijkstra and published in 1959 (Dijkstra, 1959), the Dijkstra's algorithm solves the problem of finding the shortest paths in a directed weighted graph $G = (V, E)$, where all the edge weights are non-negative. Its time complexity is $O(log(|V| * |E|))$.

Let S be a set of vertices whose final weights of the shortest paths from the starting vertice have already determined. Let C be a vector (data structure) where the index represents a vertice. The value in vector C, $C[v]$, represents the path cost to reach v from the starting vertice. Initially, S has only the starting vertice. As Dijkstra's algorithm is a greedy algorithm, at every iteration, a vertice $v_i \in V - S$, whose distance to starting vertice is as small as possible, is added to S. The algorithm stops when all the vertices are in the set S. The basic steps of the algorithm can be described as follows:

1. Add the starting vertice, v_s to set S and set its cost to zero, $C[v_s] = 0$;
2. Set to infinity the cost to reach all other vertices in the set $V - S$;
3. Initially, consider v_s as the current vertice, v_c;
4. Verify in the set $V - S$ if the neighbors of the current vertice $v_c \in S$ have cost $C[v_i]$ greater than the sum of the cost to reach the current vertice, $C[v_c]$, plus the weight of the edge connecting them, w. If so, update the cost to reach the vertice v_i as $C[v_i] = C[v_c] + w$;
5. Add to set S the vertice $v_i \in V - S$ that has the smallest cost in vector C. This is the next current vertice, v_c. Repeat the step 4 until $V - S = \emptyset$;
6. When all the vertices are added to set S, the vector C represents the cost of all the shortest paths from the starting vertice to any vertice in the graph G.

3. Proposed signature

3.1. Texture as a graph

We aim to characterize the texture by exploring the shortest paths between selected points of the image. This is similar as a traveler exploring the shortest ways of a landscape. For this purpose, the first step is to create an undirected graph that represents the neighborhood relation of an input image. In such scheme, we build a graph $G = (V, E)$ by considering each pixel $I(x, y), x = 1 \ldots M$ and $y = 1 \ldots N$ as a vertice $v \in V$ of the graph G. Each vertice is associated to an image pixel. An undirected edge $e \in E$ connects two vertices only if the Chebyshev distance between them is no longer than a value r_d. In this paper, we use $r_d = 1$:

$$E = \{e = (v, v') \in V \times V | \max(|x - x'|, |y - y'|) = 1\}. \tag{1}$$

To each edge $e \in E$ we associate a weight $w(e)$, which is defined as

$$w(e) = |I(x, y) - I(x', y')| + \frac{I(x, y) + I(x', y')}{2}, \tag{2}$$

where $I(x, y)$ and $I(x', y')$ are the gray-level intensities of the neighboring pixels associated to vertices v and v'. The expression $|I(x, y) - I(x', y')|$ measures the cost of going from one pixel to another, while the expression $(I(x, y) + I(x', y'))/2$ measures the mean altitude where this transition occurs. This calculus of mean intensity is necessary for distinguishing classes of textures that present similar patterns, but have different average gray-level intensities.

3.2. Shortest path signature

In order to characterize a texture modeled as a graph, we propose to use shortest paths computed using the Dijkstra's algorithm. To accomplish such a task, we consider the shortest paths between four sets of starting and ending vertices of the graph. These sets of vertices are the diagonal points of the texture (diagonal paths or p_{45° and p_{135°) and paths between the middle vertices of the left and right sides (horizontal path or p_{0°) and between the middle vertices of the upper and lower sides (vertical path or p_{90°), as

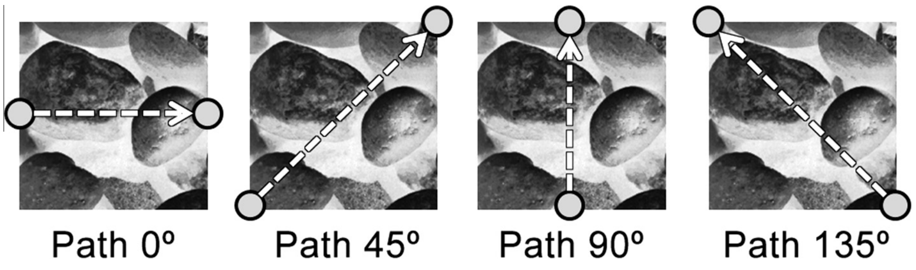

Fig. 1. Example of the four sets of starting and ending vertices considered in the calculus of the shortest paths.

shown in Fig. 1. Adopting this strategy, it is possible to measure how distinct some texture patterns are from a plane zone (an image in which all the pixels has the same gray-levels) at different orientations. Thus, this approach gives a signature of the texture image.

However, four shortest paths are not enough to capture the essence of the texture pattern. We are just having a good impression of texture's global characteristics. Locally, no measurement has been performed. Thus, it is extremely important to provide context information about pixel surroundings, which refers to a local texture analysis. We perform this by dividing the original texture into boxes of size $r \times r$, where r is a divisor of the original texture size. Then, for each box, we find the shortest paths between the four sets of vertices proposed. It is important to stress that we are using non-overlapping boxes, so that no part of a minimum path from a determined box belongs to another box. We adopt this strategy because overlapping boxes increase the computational cost (especially for larger box sizes) without a relevant improvement in the performance of the method.

To characterize a texture pattern by local and global shortest paths, we propose some feature vectors. The first two feature vectors represent a texture pattern covered by boxes of size $r \times r$. For each box, the four shortest paths $(p_{0^\circ}, p_{45^\circ}, p_{90^\circ}, p_{135^\circ})$ are obtained. Then, we compute the average and the standard deviation of each path direction to compose the following feature vectors, $\vec{\alpha}_r$ and $\vec{\beta}_r$:

$$\vec{\alpha}_r = \left[\mu_{0^\circ}, \mu_{45^\circ}, \mu_{90^\circ}, \mu_{135^\circ}\right] \tag{3}$$

and

$$\vec{\beta}_r = [\sigma_{0^\circ}, \sigma_{45^\circ}, \sigma_{90^\circ}, \sigma_{135^\circ}]. \tag{4}$$

where μ_{d° and σ_{d° represent, respectively, the average and the standard deviation of all paths in the direction d from all windows which have the same size (for instance, we can cover an image 200×200 pixel size with $2,500$ disjoint windows 4×4 pixels - each window with four directions. So, μ_{45° e σ_{45° represent, respectively, the mean and standard deviation of the $2,500$ shortest paths in a direction of 45°). These two feature vectors can be combined into a third feature vector, $\vec{\psi}_r$, which is just the concatenation of the two previous feature vectors:

$$\vec{\psi}_r = \left[\vec{\alpha}_r, \vec{\beta}_r\right]. \tag{5}$$

We also propose feature vectors which are capable of analyzing the texture image at different box sizes. This task is accomplished by the concatenation of the previous feature vectors computed for different r values:

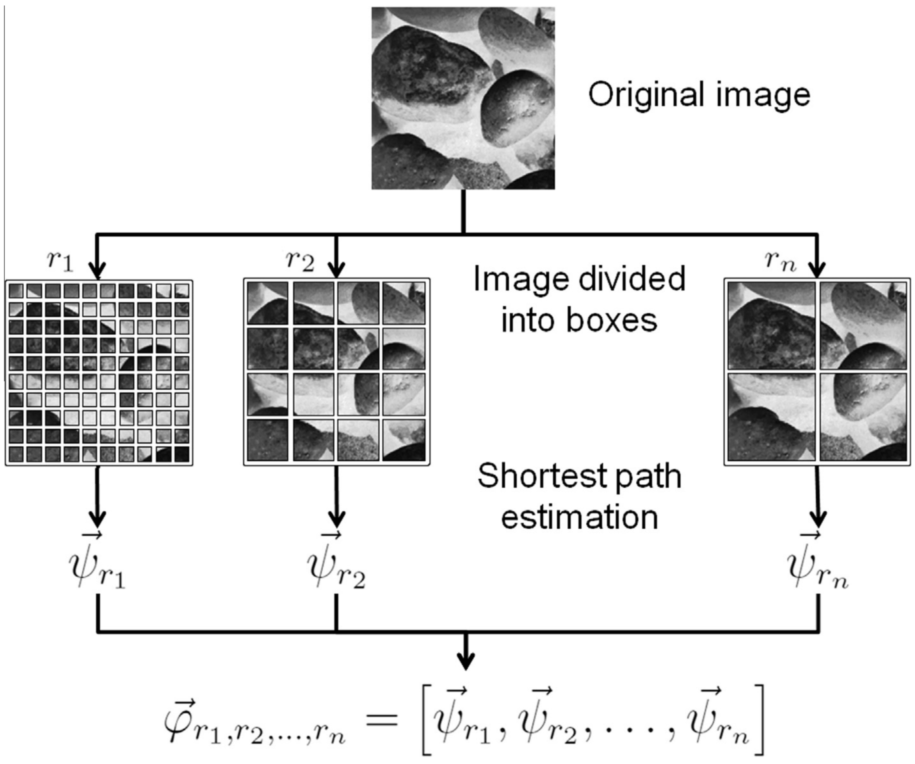

Fig. 2. Overview of the process of feature vector extraction, where r_n is the size of squares in which the image is divided into.

$$\vec{\upsilon}_{r_1,r_2,\ldots,r_n} = \left[\vec{\alpha}_{r_1}, \vec{\alpha}_{r_2}, \ldots, \vec{\alpha}_{r_n}\right], \quad (6)$$

$$\vec{\omega}_{r_1,r_2,\ldots,r_n} = \left[\vec{\beta}_{r_1}, \vec{\beta}_{r_2}, \ldots, \vec{\beta}_{r_n}\right], \quad (7)$$

$$\vec{\varphi}_{r_1,r_2,\ldots,r_n} = \left[\vec{\psi}_{r_1}, \vec{\psi}_{r_2}, \ldots, \vec{\psi}_{r_n}\right]. \quad (8)$$

To compose these feature vectors, it is necessary to divide the image into boxes of different sizes. By doing this, we aim to study how the shortest paths of the texture change when we explore the image at different scales. Fig. 2 illustrates one of these three feature vectors.

4. Experiments

To evaluate the proposed signatures of the method, we use the statistical classifier Linear Discriminant Analysis (LDA) in a leave-one-out cross validation scheme. The LDA method considers that all the classes have the same covariance matrix and separates them by hyperplanes. Details of implementation and additional information about the LDA method can be found in Everitt and Dunn (2001); Fukunaga et al. (1990). The leave-one-out cross-validation is a common approach to estimate the classifier error. It separates the data so that one sample is used as validation data and the remaining samples are used as training data. This procedure is repeated until all samples are used as validation data. The total number of errors leads to the estimation of the classification error probability.

We consider two databases to evaluate the proposed approach. The first consists of a set of textures extracted from the Brodatz book (Brodatz, 1966). This database is a well established benchmark for researches of texture analysis. For this work, we use 40 classes (D4, D5, D7, D12, D13, D15, D16, D31, D32, D33, D37, D40, D47, D49, D50, D51, D63, D71, D73, D83, D84, D85, D88, D89, D90, D92, D93, D94, D95, D96, D98, D99, D100, D103, D104, D105, D106, D108, D111, D112) and 10 images per class. Each image is 200×200 pixels with 256 gray-levels.

The second database is a more challenging texture set, because images were obtained from different viewpoints, with perspective distortions and non-rigid transformations (Lazebnik et al., 2005). This database is a collection of $1,000$ grayscale texture images (25 classes, with 40 samples each) and each image is 640×480 pixels with 256 gray-levels. In order to keep consistency with the Brodatz database, we cropped a 200×200 pixels window from the upper-left side of each image for building a new database.

We also compare the proposed approach with traditional texture analysis methods. For this comparison, we consider the following methods:

Fourier descriptors: The bi-dimensional Fourier transform is obtained from an input image. Next, a *shifting* operator is applied over the spectrum. A total of 99 descriptors is computed from this shifted image. Each descriptor is a sum of all the absolute values (Fourier spectrum) of the coefficients placed at the same radial distance from the image center (Azencott et al., 1997).

Co-occurrence matrices: They are matrices that represent a joint probability of a pair of pixels to be separated by determined distance d and direction θ. Distances of 1 and 2 with angles of $0^\circ, 45^\circ, 90^\circ, 135^\circ$, in a non-symmetric version are used for reducing the computational cost. For each co-occurrence matrix, we compute energy and entropy descriptors to compose the image feature vector because they are the most used in literature (Haralick, 1979).

Gabor filters: A Gabor filter is, basically, a Gaussian function modulated by a sinusoid oriented by a frequency and a direction. We compute 48 descriptors from the convolution of Gabor filters over an input image. A total of 24 filters (6 rotation filters and 4 scale filters), with frequency ranging from 0.05 to 0.4, are used (Manjunath and Ma, 1996).

Wavelets descriptors (Daubechies, 1992; Chang and Kuo, 1993; Randen and Husøy, 1999; Jin et al., 2011): we use the multilevel 2D wavelet decomposition in the experiment. For a given input image, we perform three dyadic decompositions using daubechies 4. Then, we compute the energy and entropy for horizontal, diagonal and vertical details, totaling 18 features (Laine and Fan, 1993).

Tourist walk: this approach considers each pixel as a tourist wishing to visit cities (other pixels) according to the rule of going to the nearest (or farthest) city that has not been visited in the last μ time steps. For a given image, we compute the tourist walk for the minimum and maximum distance rule, using time steps $\mu = \{0, 1, 2, 3, 4, 5\}$. A total of 48 descriptors for each walking histogram are computed (Backes et al., 2010).

Gravitational system: this approach analyzes texture based on representing states of a gravitational collapse process from an image and extracting information from each state using fractal dimension and lacunarity. The measure of fractal dimension is obtained by the Bouligand-Minkowski method using time steps $t = \{1, 5, 10, 15\}$ and radius values $r = \{3, 4, 5, 6, 7\}$ (Sá Junior and Backes, 2012). The measure of lacunarity is obtained by the gliding-box method using $t = \{1, 6, 12, 18\}$ and window sizes $l = \{2, 3, \ldots, 11\}$ (Sá Junior et al., 2012).

5. Results

In order to efficiently classify the texture databases, it is necessary to establish the best parameters values of the proposed method. One of these parameters is the set of box sizes r used to divide the image. We opt to use the box sizes that are divisors of the original image size (200×200 pixels). By using the divisors of the image size as box sizes, we assure that the whole image is covered by a grid of square boxes, i.e., we do not have only a box half filled with image data. Moreover, we do not consider box sizes $r = 2$ and $r = 3$, as these values do not provide meaningful information about the image pattern. We also do not consider the size $r = 200$ due to the lack of information of the standard deviation.

Tables 1 and 2 show the success rate achieved for different box sizes r in both databases. We define the success rate as the percentage of correctly classified texture samples. Despite of some oscillations, we note that the smallest box sizes lead to the best results. This is corroborated by the fact that the success rate often decreases as the box size increases, for the three feature vectors ($\vec{\alpha}_r, \vec{\beta}_r$ and $\vec{\psi}_r$). This result is explained by the fact that a small box implies in an image covered by a grid containing more squares (e.g., a box size $r = 4$ produces a grid containing $2,500$ squares)

Table 1
Success rate (%) of the method on the Brodatz database for different sizes r.

r	Success rate (%)								
	4	5	8	10	20	25	40	50	100
$\vec{\alpha}_r$	82.50	62.50	77.75	84.50	79.25	71.50	72.50	66.75	55.25
$\vec{\beta}_r$	78.25	76.75	69.50	67.00	49.25	44.00	33.25	30.25	14.00
$\vec{\psi}_r$	94.25	88.50	91.75	92.25	89.00	85.50	85.50	79.50	64.25

Table 2
Success rate (%) of the method on the UIUC database for different sizes r.

r	Success rate (%)								
	4	5	8	10	20	25	40	50	100
$\vec{\alpha}_r$	27.20	27.60	26.80	27.20	26.90	24.60	24.40	23.80	22.90
$\vec{\beta}_r$	30.90	29.30	24.50	23.60	25.60	22.40	21.40	17.00	13.70
$\vec{\psi}_r$	48.30	47.70	43.10	43.20	43.10	41.50	38.70	38.50	30.90

and, therefore, a more precise information about the texture pattern.

These results also show that the average path, $\vec{\alpha}_r$, is more representative than the standard deviation $\vec{\beta}_r$, especially in the Brodatz database. The concatenation of both measurements increases the success rate of the proposed method. This is due to the fact that while the average path can distinguish among textures with different patterns and/or small variation over the whole pattern, the standard deviation enable us to characterize textures which result in a same average path, but which present a large variance along the pattern.

In Table 3, we show the success rate achieved for both databases when we use a set of different box sizes to compute the proposed feature vectors. Results show that the success rate is better when we consider more box sizes into the feature vector. However, there is a limit of the box size which should be considered. The addition of boxes extremely large (such as $r \geqslant 50$) compromises the capability of the method to discriminate a texture pattern. An explanation for such result may be in the fact that the standard deviation provided by these box sizes is little informative.

As previously discussed in Tables 1 and 2, here we also note that the concatenation of both measurements, average and standard deviation paths, in a single feature vector, $\vec{\varphi}$, increases the success rate of the proposed method. The method reaches its best results when we consider both average and standard deviation paths ($\vec{\varphi}$ feature vector), for the box sizes $r = \{4, 5, 8, 10, 20, 25, 40\}$.

We also compare our proposed approach with traditional texture analysis methods. In this experiment, we consider the configuration of our method that achieves the best results in Table 3 ($\vec{\varphi}$ feature vector with box sizes $r = \{4, 5, 8, 10, 20, 25, 40\}$). The methods compared are configured according to either their respective papers or the common use in literature. Table 4 presents the results of this comparison. Results indicate that our approach achieves the best results in both experiments, overcoming all compared methods, except for the gravitational system (gravitational system with Bouligand–Minkowski) method in Brodatz database (difference of 0.25%, that is, only one sample). It is important to emphasize that our approach is 1.5% superior to the third best methods (Gabor filters and gravitational system with gliding-box) in the Brodatz database. This is a relevant difference because the three methods are close to 100.00% of success rate. On the other hand, despite the lower success rate, Gabor filters use less descriptors (48) than our approach (56). Our approach uses eight more descriptors to increase its success rate in 1.50%. This clearly sounds as a disadvantage of our method. However, it is important to recall the results achieved for the $\vec{\varphi}$ feature vector with box sizes $r = \{4, 5, 8, 10, 20\}$. This configuration uses only 40 descriptors and presents a success rate (97.50%) that still overcomes the results achieved by the Gabor filters.

When we consider the UIUC database, it is clear the efficiency of our approach, as it achieved a result 9.50% superior to the second best method (gravitational system with Bouligand-Minkowski), i.e., more 95 images are correctly classified. This superior result is achieved because the proposed method is quite robust to rotation, once if an image is rotated in multiples of 45°, the same shortest paths are obtained - even if an image is rotated in other angles, it is unlikely that the shortest paths suffer significant changes. Moreover, the method can cope with images taken from different viewpoints (same distance from the image), provided that they differ in small angles, because images that satisfy this condition present similar generated graphs.

Because Gabor filters obtain a high performance and also depend on scale and orientation, we made an additional experiment with this method using the same values for orientation (0°, 45°, 90° and 135°) employed in our proposed approach and several values for scale parameter in order to better evaluate the performance of the SPG method. Table 5 presents the results of this comparison. The results show that the success rate decreases when we use greater values for scale parameter. This differs from the behavior of our method, which obtains its best performance with seven scale values. Moreover, all the success rates obtained by Gabor filters are smaller than the success rate of the SPG method. These results confirm the efficiency of our proposed approach for texture discrimination.

Finally, we perform two additional experiments with the original images from UIUC database (640×480 pixels), presented in Lazebnik et al. (2005). In this reference the authors obtained 96.00% of correct classification using only 20 samples of each class for the training set (KNN classifier with EMD distance). The first experiment employs this same configuration for classification and a feature vector with window sizes $r = \{4, 5, 8, 20, 30, 40\}$, which yields 52.86% of success rate. This low performance can

Table 3
Success rate (%) of the method on the Brodatz and UIUC databases for different sets of sizes r.

Sets of window sizes $\{r_1, r_2, \ldots, r_n\}$	$\vec{\upsilon}_{r_1,r_2,\ldots,r_n}$	$\vec{\omega}_{r_1,r_2,\ldots,r_n}$	$\vec{\varphi}_{r_1,r_2,\ldots,r_n}$
Brodatz database	*Success rate* (%)		
{4, 5}	82.25	90.25	94.25
{4, 5, 8}	91.50	92.00	96.25
{4, 5, 8, 10}	92.00	92.25	96.75
{4, 5, 8, 10, 20}	95.00	94.25	97.50
{4, 5, 8, 10, 20, 25}	95.00	92.75	97.25
{4, 5, 8, 10, 20, 25, 40}	94.75	92.50	98.50
{4, 5, 8, 10, 20, 25, 40, 50}	94.00	92.00	97.75
{4, 5, 8, 10, 20, 25, 40, 50, 100}	95.50	91.75	97.75
UIUC database	*Success rate* (%)		
{4, 5}	36.80	41.90	57.70
{4, 5, 8}	44.60	49.00	63.00
{4, 5, 8, 10}	46.60	49.10	64.20
{4, 5, 8, 10, 20}	51.50	52.80	65.80
{4, 5, 8, 10, 20, 25}	52.20	50.50	65.40
{4, 5, 8, 10, 20, 25, 40}	55.10	50.80	67.30
{4, 5, 8, 10, 20, 25, 40, 50}	54.40	49.70	66.40
{4, 5, 8, 10, 20, 25, 40, 50, 100}	54.10	49.60	66.00

Table 4
Comparison results for different texture methods in the Brodatz and UIUC databases.

Method	Success rate (%)	
	Brodatz	UIUC
Fourier descriptors	87.75	35.10
Co-occurrence matrices	82.50	41.10
Gabor filters	97.00	56.10
Wavelet descriptors	87.50	38.80
Tourist walk	95.50	48.70
Gravitation (Bouligand–Minkowski)	98.75	57.80
Gravitation (gliding-box)	97.00	54.10
Proposed approach	98.50	67.30

Table 5
Comparison results for the proposed method and different configurations of Gabor filters in the Brodatz and UIUC databases.

Method	Success rate (%)	
	Brodatz	UIUC
Gabor filters (4 scales + 4 rotations)	97.50	54.60
Gabor filters (5 scales + 4 rotations)	96.50	53.60
Gabor filters (6 scales + 4 rotations)	95.50	52.20
Gabor filters (7 scales + 4 rotations)	94.25	51.10
Gabor filters (8 scales + 4 rotations)	94.50	49.30
Gabor filters (9 scales + 4 rotations)	94.50	46.90
Gabor filters (10 scales + 4 rotations)	94.50	47.30
Proposed approach	98.50	67.30

be explained by the fact that features provided by the SPG method are strongly correlated and, therefore, cannot be adequately analyzed by simpler classifiers, such as KNN.

On the other hand, a second experiment using the same window size set and LDA method with leave-one-out scheme yields 88.00% of success rate. This result is 20.70% superior to the success rate obtained with the images from UIUC database 200×200 pixels (67.30%). We achieve this higher success rate for the original 640×480 images because they are more representative of the texture pattern. For instance, because the original images were extracted from different viewpoints, in many cases the image 200×200 cropped from their upper-left corner are blurred or distorted when compared to the whole image.

Finally, even though this result (88.00%) is smaller than the success rate presented in Lazebnik et al. (2005) (96.00%), it is very significant because the images from UIUC database were taken from very different viewpoints, with big distortions of perspective and non-rigid transformations. For such hard images, methods based on texture primitives (for instance, the method presented in Lazebnik et al. (2005)) have advantages over the SPG method, which is based on statistical moments.

6. Conclusion

This work presents a novel method for extracting texture information. Given an input image, we convert it into an undirected weighted graph whose weights are defined by the image gray-levels. Then, we compute shortest paths between different square regions of the graph/image. We start with large square which are diminished at each step, in a multi-scale approach. This enable us to perform a texture analysis which considers both micro and macro texture information.

The proposed approach shows superior results to those yielded by classical and novel methods, especially when tested on a UIUC database. We achieve the best results when a specific set of box sizes is used, and realize that the addition of boxes extremely large (such as $r \geqslant 50$) compromises the capability of the method to discriminate a texture pattern. Thus, this work opens a promising source of research, which offers significant information from textures and, therefore, becomes an important tool for dealing with image analysis problems.

References

Azencott, R., Wang, J.-P., Younes, L., 1997. Texture classification using windowed fourier filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2), 148–153.

Backes, A.R., Casanova, D., Bruno, O.M., 2009. Plant leaf identification based on volumetric fractal dimension. IJPRAI 23 (6), 1145–1160.

Backes, A.R., Gonçalves, W.N., Martinez, A.S., Bruno, O.M., 2010. Texture analysis and classification using deterministic tourist walk. Pattern Recognition 43 (3), 685–694.

Beck, J., Sutter, A., Ivry, R., 1987. Spatial frequency channels and perceptual grouping in texture segregation. Computer Vision, Graphics and Image Processing 37, 299–325.

Bellman, R., 1958. On a routing problem. Quarterly of Applied Mathematics 16 (1), 87–90.

Brodatz, P., 1966. Textures: A Photographic Album for Artists and Designers. Dover Publications, New York.

Casanova, D., Sá Junior, J.J.M., Bruno, O.M., 2009. Plant leaf identification using gabor wavelets. International Journal of Imaging Systems and Technology 19 (1), 236–243.

Chang, T., Kuo, C.-C.J., 1993. Texture analysis and classification with tree-structure wavelet transform. IEEE Transactions on Image Processing 2 (4), 429–441.

Chen, Y.Q., Bi, G., 1999. On texture classification using fractal dimension. IJPRAI 13 (6), 929–943.

Costa, L.F., Rodrigues, F.A., Travieso, G., Boas, P.R.V., 2007. Characterization of complex networks: a survey of measurements. Advances in Physics 56, 167–242.

Daubechies, I., 1992. Ten lectures on wavelets, SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271.

Drozdek, A., 2000. Data Structures and Algorithms in C++, 2nd Edition. Brooks/Cole Publishing Co., Pacific Grove, CA, USA.

Ebert, D., Musgrave, K., Peachey, D., Perlin, K., 1994. Worley, Texturing and Modeling: A Procedural Approach. Academic Press.

Emerson, C.W., Lam, N.N., Quattrochi, D.A., 1999. Multi-scale fractal analysis of image texture and patterns. Photogrammetric Engineering and Remote Sensing 65 (1), 51–62.

Euler, L., 1736. Solutio problematis ad geometriam situs pertinentis. Commentarii Academiae Scientiarum Imperialis Petropolitanae 8, 128–140.

Everitt, B.S., Dunn, G., 2001. Applied Multivariate Analysis, Arnold.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, Academic Press.

Haralick, R.M., 1979. Statistical and structural approaches to texture. Proc. IEEE 67 (5), 786–804.

Jagannathan, A., Miller, E.L., 2002. A graph-theoretic approach to multiscale texture segmentation. IEEE International Conference on Image Processing 2, 777–780.

Jin, X., Gupta, S., Mukherjee, K., Ray, A., 2011. Wavelet-based feature extraction using probabilistic finite state automata for pattern classification. Pattern Recognition 44 (7), 1343–1356.

Jirik, M., Ryba, T., Zelezny, M., 2011. Texture based segmentation using graph cut and Gabor filters. Pattern Recognition and Image Analysis 21 (2), 258–261.

Julesz, B., 1975. Experiments in the visual perception of texture. Scientific American 232 (4), 34–43.

Kaplan, L.M., 1999. Extended fractal analysis for texture classification and segmentation. IEEE Transactions on Image Processing 8 (11), 1572–1585.

Laine, A., Fan, J., 1993. Texture Classification by Wavelet Packet Signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (11), 1186–1191.

Lazebnik, S., Schmid, C., Ponce, J., 2005. A sparse texture representation using local affine regions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8), 1265–1278.

Lu, C.S., Chung, P.C., Chen, C.F., 1997. Unsupervised texture segmentation via wavelet transform. Pattern Recognition 30 (5), 729–742.

Manjunath, B.S., Ma, W.-Y., 1996. Texture features for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8), 837–842.

Murino, V., Ottonello, C., Pagnan, S., 1998. Noisy texture classification: a higher order statistics approach. Pattern Recognition 31 (4), 383–393.

Randen, T., Husøy, J.H., 1999. Filtering for texture classification: a comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4), 291–310.

Sá Junior, J.J.M., Backes, A.R., 2011. A Simplified Gravitational Model for Texture Analysis. In: Lecture Notes in Computer Science. In: Real, P., Dıaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W.G. (Eds.), vol. 6854. Springer, pp. 26–33.

Sá Junior, J.J.M., Backes, A.R., 2012. A simplified gravitational model to analyze texture roughness. Pattern Recognition 45 (2), 732–741.

Sá, J.J.M., Jr., Backes, A.R., Cortez, P.C., in press. A simplified gravitational model for texture analysis. Journal of Mathematical Imaging and Vision. http://dx.doi.org/10.1007/s10851-012-0408-1.

Sengür, A., Türkoglu, I., Ince, M.C., 2007. Wavelet packet neural networks for texture classification. Expert System Applications 32 (2), 527–533.

Tricot, C., 1995. Curves and Fractal Dimension. Springer-Verlag.

Xu, C.L., Xen, Y.Q., 2004. Statistical landscape features for texture classification. IEEE International Conference on Pattern Recognition 1, 676–679.