



Texture descriptor combining fractal dimension and artificial crawlers

Wesley Nunes Gonçalves^{a,b}, Bruno Brandoli Machado^{a,b},
Odemir Martinez Bruno^{b,*}

^a Federal University of Mato Grosso do Sul, Rua Itibiré Vieira, s/n, CEP 79907-414, Ponta Porã - MS, Brazil

^b Scientific Computing Group¹, São Carlos Institute of Physics, University of São Paulo, Caixa Postal 369 CEP: 13560-970 São Carlos - SP, Brazil

HIGHLIGHTS

- A hybrid fractal–swarm method for texture analysis is proposed.
- It performs artificial crawlers over the image considering two rules: minimum and maximum.
- The fractal dimension is applied to the swarm map of states, carrying out a texture descriptor.

ARTICLE INFO

Article history:

Received 7 April 2013

Received in revised form 3 October 2013

Available online 16 October 2013

Keywords:

Fractal dimension

Artificial crawler

Texture analysis

ABSTRACT

Texture is an important visual attribute used to describe images. There are many methods available for texture analysis. However, they do not capture the detail richness of the image surface. In this paper, we propose a new method to describe textures using the artificial crawler model. This model assumes that agents can interact with the environment and each other. Since this swarm system alone does not achieve a good discrimination, we developed a new method to increase the discriminatory power of artificial crawlers, together with the fractal dimension theory. Here, we estimated the fractal dimension by the Bouligand–Minkowski method due to its precision in quantifying structural properties of images. We validate our method on two texture datasets and the experimental results reveal that our method leads to highly discriminative textural features. The results indicate that our method can be used in different texture applications.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The discrimination of visual texture has played an important role in computer vision and image analysis. Although the ability of texture discrimination is apparently easy for human beings, the description by using texture methods has proven to be very complex. Several methods have been proposed to characterize texture images. They are based on *statistical analysis of the spatial distribution* (e.g., co-occurrence matrices [1,2], local binary pattern [3] and interaction map [4]), *stochastic models* (e.g., Markov random fields [5,6]), *spectral analysis* (e.g., Fourier descriptors [7], Gabor filters [8,9] and wavelet transform [10,11]), *structural models* (e.g., mathematical morphology [12] and geometrical analysis [13]), and *complexity analysis* (e.g., fractal dimension [14,15]). Despite the fact they have thoroughly been studied, few methods are able to successfully discriminate the different texture patterns found in nature.

* Corresponding author. Tel.: +55 1633738728.

E-mail addresses: wesley.goncalves@ufms.br (W.N. Gonçalves), bruno.brandoli@ufms.br (B.B. Machado), bruno@ifsc.usp.br (O.M. Bruno).

¹ <http://scg.ifsc.usp.br>

Swarm systems or multi-agent systems, have been long applied in computer vision [16–21]. In texture analysis, the swarm system can be found in a select group of approaches, such as the deterministic tourist walk [22–24], the ant colony [25], and the artificial crawlers [26,27]. The basic idea of the swarm algorithms consists of creating a system by means of agent interaction, i.e., a distributed agent system with parallel processing, and autonomous computing. In this paper, we propose a novel method for texture analysis based on the artificial crawler model [26,27]. This swarm system consists of a population of agents, referred here to as artificial crawlers, that interact with each other and the environment, in this case an image. Each artificial crawler occupies a pixel, and its goal is to move to the neighbor pixel of greater intensity. The agents store their current position in the image, and a corresponding energy that can wax or wane their lifespan depending on the energy consumption of the image. The population of artificial crawlers stabilizes after a certain number of iterations, i.e., when there is no change in their spatial positions.

In the original swarm system [26,27] the artificial crawlers only move in the direction of maximum intensity, thus characterizing regions of high intensities in the image. However, in texture analysis, regions of low intensities are as important as regions of high intensities. Therefore, we propose a new rule of movement that also moves artificial crawler agents in the direction of lower intensity. Our approach differs from the original artificial crawler model in terms of movement: each agent is able to move to the higher altitudes, as well as to lower ones. To quantify the state of the swarm system after the stabilization, we propose to employ the Bouligand–Minkowski fractal dimension method [28,15]. The fractal dimension method is widely used to characterize the roughness of a surface, which is related to the physical properties. In Ref. [23], the authors have also used fractal dimension to characterize the agents. The main differences of this approach lie in the type of agents and the dilation process used to estimate the fractal dimension. First, the earlier work proposed the use of deterministic partially self-avoiding walks; the agents do not interact with each other. We, on the other hand, use artificial crawlers, which are based on agent interaction. Further, the earlier work estimates the fractal dimension of the attractors found by the agents, while we estimate the fractal dimension based on the energy information and the spatial position of each agent after the stabilization.

We have conducted experiments in two datasets widely accepted in the literature of texture analysis. Experimental results have shown that our method overcomes different state-of-the-art methods over the Vistex dataset. Besides, our approach significantly improves the classification rate compared to the original artificial crawler method. The superior results rely on two facts: the fractal dimension estimation of the swarm system and the two rules of movement. On the one hand, the use of both rules of movement characterizes both regions of the texture image. On the other hand, the fractal dimension improves the ability of discrimination obtained from the swarm system of artificial crawlers. Moreover, the idea of fractal dimension estimation can be used for other swarm systems.

The main contributions of this paper are:

- A new rule of movement for the artificial crawler method. The original method fails to describe images because it moves the agents to higher intensities only. The proposed method describes images by using two rules of movement, i.e., the swarm system finds the minima and maxima of images.
- A new methodology to image description based on the energy information acquired from two rules of movement. Although we can find the minima and maxima of images directly, the underlying idea is to characterize the path of movement during the evolution process. In this case, the energy information was considered the most important attribute due to its capacity of representing the interaction between the movement of agents and the environment.
- To enhance the discriminatory power of our method, we use the energy information and the spatial position of each agent to estimate the fractal dimension of the image surface, in this paper the Bouligand–Minkowski fractal dimension is employed.

This paper is organized as follows. In Section 2, we describe the artificial crawler model in detail. In Section 3, we present the basis for the fractal dimension and the Bouligand–Minkowski method. A new method for texture analysis based on fractal dimension of artificial crawlers is presented in Section 4. Finally, in Section 5 we report the experimental results, followed by the conclusion in Section 6.

2. Artificial crawler model

The texture method proposed in this study is based on the artificial crawler model proposed in Refs. [26,27]. Their agent-based model was first proposed in Ref. [26] and then extended in Ref. [27]. In order to describe this model, let us consider an image which consists of a pair (\mathcal{I}, I) – a finite set \mathcal{I} of pixels and a mapping I that assigns to each pixel $p = (x_p, y_p)$ in \mathcal{I} an intensity $I(p) \in [0, 255]$. Also, let us consider a neighborhood $\eta(p)$ that consists of pixels q whose Euclidean distance between p and q is smaller than or equal to $\sqrt{2}$ (eight-connected pixels):

$$\eta(p) = \{q \mid d(p, q) \leq \sqrt{2}\}$$

$$d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}.$$
(1)

In image analysis, the artificial crawler model assumes that each agent occupies one pixel of the image. At each time t , artificial crawlers $A_t^i = \{e_t^i, p_t^i\} \forall i \in [0, N]$ are characterized by two attributes. The first attribute e_t^i holds the current level of energy. Such energy can either wax or wane their lifespan according to the energy consumption and influence of

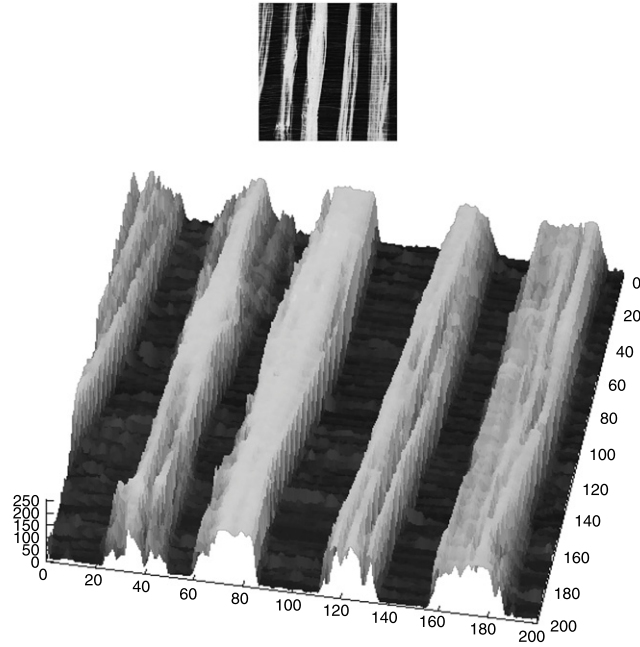


Fig. 1. The environment of the artificial crawler. At the top is shown a textured image and below its respective 3D surface.

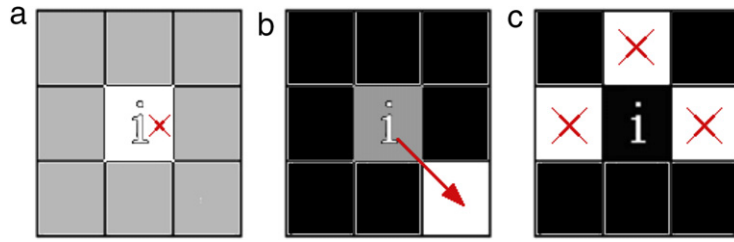


Fig. 2. Example of the three possible steps of an artificial crawler considering its eight neighbors.

the environment. The second attribute p_t^i is the current position of the artificial crawler in the image. The artificial crawlers act upon an environment. In images, the environment is mapped as a 3D surface with different altitudes that correspond to gray values on the z-axis. Higher intensity pixels supply nutrients to the artificial crawlers (increase its energy), while lower altitudes correspond to the land. Fig. 1 shows a textured image and the peaks and valleys where the artificial crawlers can increase or decrease its energy live.

The N artificial crawlers begin with equal energy e_{init} and are placed at random on the surface (pixels) of the textured image:

$$\begin{aligned} p_0^i &= \text{rand}(I) \\ e_0^i &= e_{init}. \end{aligned} \quad (2)$$

Then the evolution process starts following a set of specific rules. The aim of the artificial crawler is to move to areas of higher altitudes in order to absorb energy and sustain life. This way, the next step $p_{t+1}^i = f(p_t^i)$ depends on the gray level of its neighbors according to Eq. (3). First, the artificial crawler settles down if the gray levels of its eight neighbors are lower than itself (Fig. 2(a)). Second, the artificial crawler moves to a specific pixel if there exist one of its eight neighbors with unique higher intensity (Fig. 2(b)). Third, if there exist more than one neighbor with higher intensity, an artificial crawler moves to the pixel that has already been occupied by another artificial crawler at any time (Fig. 2(c)). Otherwise, it moves to one of the pixels randomly.

$$f(p_t^i) = \begin{cases} p_t^i, & \text{if } I(p_t^i) \geq I(p) \forall p \in \eta(p_t^i) \\ p, & \text{if } I(p) > I(p_t^i), I(p) > I(q) \forall p, q \in \eta(p_t^i), p \neq q \\ p, & \text{if } I(p) > I(p_t^i), I(p) \geq I(q) \forall p, q \in \eta(p_t^i), p \neq q, p \text{ was visited.} \end{cases} \quad (3)$$

Given the new position of the artificial crawler, the energy absorption from the environment is performed:

$$e_{t+1}^i = e_t^i + \lambda I(p_{t+1}^i) - 1 \quad (4)$$

where λ is the rate of absorption over the gray level of the current pixel $I(p_{t+1}^i)$. All artificial crawlers lose a unit of energy which means that the artificial crawler loses energy at each step if $\lambda * I(p_{t+1}^i) < 1$. For the default value of $\lambda = 0.01$, it means that the artificial crawler loses energy if it goes to a pixel whose gray level is less than 100 and gains energy otherwise. The energy is bounded by limit e_{max} , i.e. if $e_{t+1}^i > e_{max}$ then $e_{t+1}^i = e_{max}$. Also, an artificial crawler keeps living in the next generation if its energy is higher than a certain threshold e_{min} .

After the energy absorption, the law of the jungle is performed. In this law, an artificial crawler with higher energy eats up another with lower energy if they are in the same pixel, i.e. A_{t+1}^i eats up A_{t+1}^j if $p_{t+1}^i = p_{t+1}^j$, $e_{t+1}^i \geq e_{t+1}^j$, $i \neq j$. This law is inspired by nature and assumes that the artificial crawlers with higher energy are more likely to reach the peaks of the environment.

The evolution process converges to an equilibrium state when no further artificial crawlers are in movement (they are dead or settled down). In the original method, features are extracted by means of the number of artificial crawlers at each iteration and colonial properties. Each texture image is represented by four curves of evolution: (1) curve of living artificial crawlers, (2) curve of settled artificial crawlers, (3) curve of colony formation at certain radius and (4) scale distribution of colonies. This representation has two major drawbacks: (i) the vector obtained is high dimensional, which leads us to the curse of dimensionality and (ii) the extraction of this vector is very time consuming due to the colony estimation.

3. Fractal dimension

In 1977, Mandelbrot introduced a new mathematical concept to model natural phenomena, named fractal geometry [29]. This formulation received a lot of attention due to its ability to describe irregular shapes and complex objects that Euclidean geometry fails to analyze. In contrast, fractal geometry assumes that an object holds a non-integer dimension. Thus, estimating the fractal dimension of an object is basically related to its complexity. The patterns are characterized in terms of space occupation and self-similarity at different scales. The interactive construction process of the Von Koch curve is a typical example of self-similarity of fractals [14].

The first definition of dimension was proposed by the Hausdorff–Besicovitch measure [30], which provided the basis of the fractal dimension theory. He defined a dimension for point sets as a fraction greater than their topological dimension. Formally, given $X \in \mathbb{R}^d$, a geometrical set of points, the Hausdorff–Besicovitch dimension $D_H(X)$ is calculated by:

$$D_H(X) = \inf\{s : H^s(X) = 0\} = \sup\{H^s(X) = \infty\}. \quad (5)$$

where $H^s(X)$ is the s -dimensional Hausdorff measure (in Eq. (6)).

$$H^s(X) = \liminf_{\delta \rightarrow 0} \left[\sum_{i=1}^{\infty} |U_i|^s : U_i \text{ is a } \delta\text{-cover of } X \right] \quad (6)$$

where $|\cdot|$ stands for the diameter in \mathbb{R}^d , i.e. $|U| = \sup |x - y| : x, y \in U$.

In image analysis, the use of the Hausdorff–Besicovitch definition may be impracticable [31]. An alternative definition generalized from the topological dimension is commonly used. According to this definition, the fractal dimension D of an object X is:

$$D(X) = \lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log \frac{1}{\epsilon}} \quad (7)$$

where $N(\epsilon)$ stands for the number of objects of linear size ϵ needed to cover the whole object X .

There are a lot of algorithms to estimate the fractal dimension of objects or surfaces. The most well known algorithms are: box-counting [32], differential box-counting [33], ϵ -blanket [34], fractal model based on fractional Brownian motion [35], power spectrum method [35], Bouligand–Minkowski [28] among others; as well as extensions of fractals, such as multi-fractals [36], multi-scale fractals [15] and fractal descriptors [37–40]. One of the most accurate methods to estimate the fractal dimension is the Bouligand–Minkowski method [28,15,41]. The Bouligand–Minkowski fractal dimension $D_B(X)$ depends on a symmetrical structuring element Y :

$$D_B(X, Y) = \inf\{\lambda, m_B(X, Y, \lambda) = 0\} \\ m_B(X, Y, \lambda) = \lim_{\epsilon \rightarrow 0} \frac{V(\partial X \oplus \epsilon Y)}{\epsilon^{n-\lambda}} \quad (8)$$

where m_B is the Bouligand–Minkowski measure, ϵ is the radius of the element Y and V is the volume of the dilation between element Y and boundary ∂X . To eliminate the explicit dependence on the element Y , a simplified version of the Bouligand–Minkowski fractal dimension can be described by using neighborhood techniques as:

$$D_B(X) = \lim_{\epsilon \rightarrow 0} \left(D_T - \frac{\log V(X \oplus Y_\epsilon)}{\log \epsilon} \right). \quad (9)$$

For instance considering an object $X \in \mathbb{R}^3$, the topological dimension $D_T = 3$ and Y_ϵ is a sphere of diameter ϵ . Varying the radius ϵ , it estimates the fractal dimension based on the size of the influence area V created by the dilation of X by Y_ϵ .

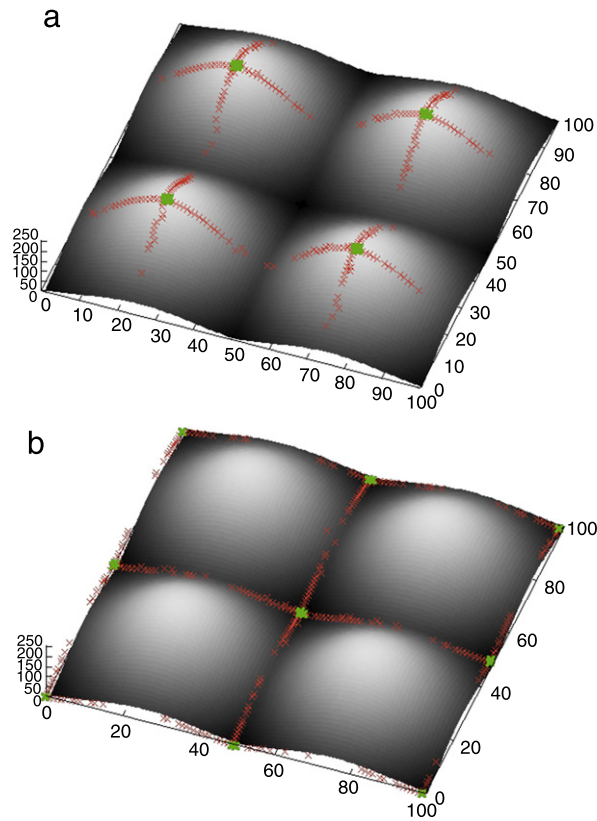


Fig. 3. The final position of 1000 artificial crawlers, (a) by using the rule of movement *max* and (b) by using the rule of movement *min*. Green marks stand for live artificial crawlers while red marks represent dead artificial crawlers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Proposed method

In this section, we describe the proposed method, which is based on the fractal dimension of artificial crawlers. Basically, our method can be divided into two parts: artificial crawlers are performed in the texture image and then the fractal dimension of these artificial crawlers is estimated. The next sections describe these steps of our method.

4.1. Artificial crawler model in images

Although the original artificial crawler method achieves promising results, the idea of moving to pixels with higher intensities does not extract all the richness of textural pattern of the images. In the method proposed here, the independent artificial crawlers are also able to move to lower intensities (valleys). It allows the model to take full advantage and capture the richness of details present in peaks and valleys of the images.

In the first step, the artificial crawlers move to higher intensities as in the original method. Thus, artificial crawlers $A_T^i = \{p_T^i, e_T^i\}$ are obtained after the evolution process converges, where T is the number of steps needed till the system converges. Throughout the paper, the artificial crawlers which move to higher intensities will be referred to as $U_T^i = \{p_T^i, e_T^i\}$ and this rule of movement will be referred to as *max*. Fig. 3 shows an example of 1000 artificial crawlers using the rule of movement *max* and *min*. Although the image in Fig. 3(a) is an elaborate example we present how the agents find the maxima accordingly the *max* rule. The green marks stand for the final position (convergence) of the live artificial crawlers while the red ones represent the final position of the dead artificial crawlers. As we can see, the live artificial crawlers can achieve the highest intensities. As important as the live artificial crawlers, the dead ones aggregate information from the surface of the environment.

Artificial crawlers are born in different areas of altitudes and their migration activity either leads to gain or loss of energy. The energy of each agent is directly influenced by the absorption of the environment. For instance, let us suppose that the environment has two peaks, p_1 and p_2 , with different altitudes, a_{p_1} and a_{p_2} . If $a_{p_1} > a_{p_2}$, the energy of the agent that reached the peak p_1 is higher than the energy of another agent that reached the peak p_2 . It occurs because the energy absorption is higher for the agent that is climbing the peak p_1 . Therefore, we can say that the energy corresponds to the history of the agents' steps.

In Fig. 3(a), we can observe that the original method only describes the peaks of a texture image. In addition, we propose to move artificial crawlers towards lower intensities. In this approach, artificial crawlers $Q_t^i = \{p_t^i, e_t^i\}$ are randomly placed in the image with initial energy e_{init} . Then, the evolution process is modified so that the next step of an artificial crawler is to move towards the lower intensity (Eq. (10)). This rule of movement will be referred to throughout the paper as *min*.

$$f(p_t^i) = \begin{cases} p_t^i, & \text{if } I(p_t^i) \leq I(p) \forall p \in \eta(p_t^i) \\ p, & \text{if } I(p) < I(p_t^i), I(p) < I(q) \forall p, q \in \eta(p_t^i), p \neq q \\ p, & \text{if } I(p) < I(p_t^i), I(p) \leq I(q) \forall p, q \in \eta(p_t^i), p \neq q, p \text{ was visited.} \end{cases} \quad (10)$$

An example of the artificial crawlers using the rule of movement *min* can be seen in Fig. 3(b). Again, green marks represent the final position of live artificial crawlers while red marks represent the dead artificial crawlers. These artificial crawlers complement the artificial crawlers that use the rule of movement *max*, aggregating more information about the surface.

At the end of this step, we have two populations of N artificial crawlers $U_T^i = \{p_T^i, e_T^i\}$ and $Q_T^i = \{p_T^i, e_T^i\}$ which correspond to the artificial crawlers using rules of movement *max* and *min*, respectively.

4.2. Fractal dimension of artificial crawlers

In this section, we describe how to quantify the population of artificial crawlers using the fractal dimension theory. To estimate the fractal dimension using the Bouligand–Minkowski method, the population of artificial crawlers can be easily mapped onto a surface $S \in \mathbb{R}^3$, by converting the position $p_T^i = \{x_i, y_i\}$ and the energy e_T^i of each artificial crawler into a 3D point $s_i = (x_i, y_i, e_T^i)$. The energy is important because it contains the information related to the evolution process of the artificial crawlers. This mapping can be seen in Fig. 4(a). We should note that the Z axis is the energy of the artificial crawlers.

The Bouligand–Minkowski method estimates the fractal dimension based on the size of the influence area $|S(r)|$ created by the dilation of S by a radius r . Thus varying the radius r , the fractal dimension of surface S is given by:

$$D = 3 - \lim_{r \rightarrow 0} \frac{\log V(r)}{\log r} \quad (11)$$

where $V(r)$ is the influence volume obtained through the dilation process of each point of S using a sphere of radius r :

$$V(r) = |\{s' \in \mathbb{R}^3 \mid \exists s \in S : |s - s'| \leq r\}|. \quad (12)$$

The dilation process is illustrated in Fig. 4. A group of artificial crawlers is mapped onto a 3D space, shown in Fig. 4(a). Each point of the 3D space is dilated by a sphere of radius r (Fig. 4(b) and (c)). As the value of radius r is increased, more collisions are observed among the dilated spheres. These collisions disturb the total influence volume $V(r)$, which is directly linked to the roughness of the surface.

From the linear regression of the plot of $\log r \times \log V(r)$, the Bouligand–Minkowski fractal dimension is computed by:

$$D = 3 - \alpha \quad (13)$$

where α is the slope of the estimated line.

4.3. Feature vector

Although the fractal dimension provides a robust mathematical model, it describes each object by only one real value D – the fractal dimension. Objects with distinct shapes can have the same fractal dimension, for instance, the very well known fractals: Peano curve, dragon curve, Julia set and the boundary of the Mandelbrot set have the same Hausdorff dimension equal to 2. To overcome this characteristic the concept of multi-scale fractal dimension [15] and the fractal descriptors [37] were developed. In this way, the fractal dimension of the object is considered in different scales. It provides a rich shape descriptor that can be successful to discriminate shape and patterns [15].

In order to improve the discrimination power of our method, we use the entire curve $V(r)$ instead of using only the fractal dimension:

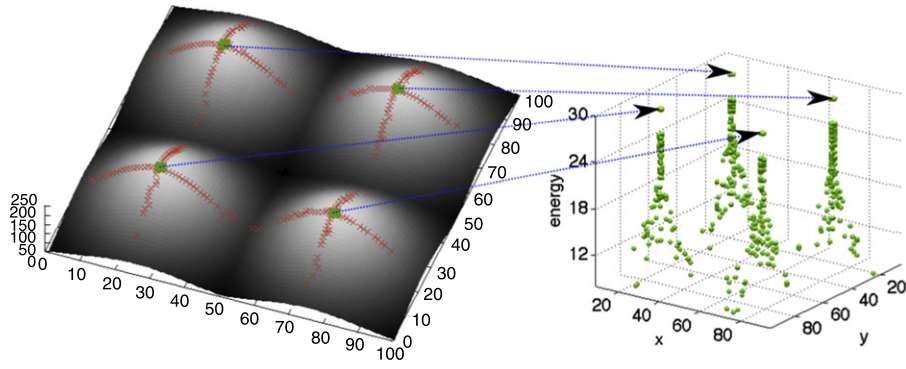
$$\varphi_\tau = [V(1), \dots, V(r_m)] \quad (14)$$

where τ is the rule of movement used by the artificial crawler and r_m is the maximum radius.

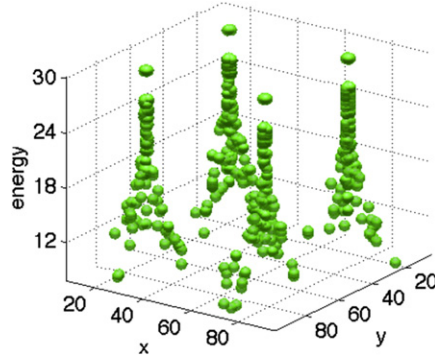
Considering that we have two rules of movement, the final feature vector is composed by the concatenation of φ_{max} and φ_{min} according to Eq. (15). The feature vectors φ_{max} and φ_{min} are obtained by using the fractal dimension estimation of artificial crawlers U_T^i and Q_T^i after the stabilization, respectively.

$$\varphi = [\varphi_{max}, \varphi_{min}]. \quad (15)$$

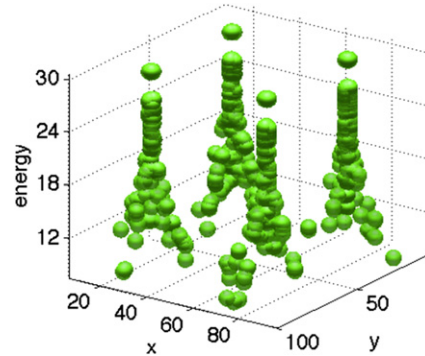
The importance of using both rules is corroborated in Fig. 5. Fig. 5(b) and (d) show the feature vectors by using φ_{max} only, and Fig. 5(c) and (e) show the feature vectors by using φ_{min} only. An example of those feature vectors are obtained for four different image classes, as shown in Fig. 5(a). For clarify, each class contains 10 samples. The classes D16 and D18 are discriminated using the rule of movement *max* (Fig. 5(b)), while the rule of movement *min* is not able to discriminate those two classes accordingly (Fig. 5(c)). On the other hand, the classes D49 and D93 are only discriminated if the rule of movement *min* is used (Fig. 5(e)). These plots corroborate the importance of using both rules of movement for texture recognition.



(a) Artificial crawlers mapped onto a 3D space by converting the final position and the energy into a point in the surface.



(b) $r = 2$.



(c) $r = 3$.

Fig. 4. An illustration of the dilation process for the fractal dimension estimation of artificial crawlers. The final position of the artificial crawlers was obtained using the rule of movement *max* and maximum energy $e_{max} = 30$.

4.4. Computational complexity

In the proposed method, N artificial crawlers are performed in the image of size $W \times W$ pixels. The swarm system converges after M steps, which leads to a computational complexity of $O(NM)$. After stabilization, we propose to quantify the swarm system by means of the fractal dimension. To calculate the dilation process, the Euclidean distance transform [42] is a powerful and efficient tool. This transform calculates the distance between each point of the 3D space and the surface. Several authors [43,44,42] proposed algorithms for computing Euclidean distance transform in linear time. The time complexity is linear in the number of points of the 3D space, which is $O(W \times W \times e_{max})$ – $W \times W$ is the size of the image and e_{max} is the maximum energy of the agents. Usually, the maximum energy e_{max} is a small number (e.g. in this work the maximum energy is 20). Thus, we can ignore e_{max} in the complexity, since $W \gg 20$ in image applications. Finally the computational complexity of the proposed method is stated as $O(NM + W^2)$.

Let us discuss the best, worst and average case based on the number of steps of the swarm system. The best case considers that the swarm system converges in one step ($M = 1$). Thus, the computation complexity is $O(N + W^2)$. In the worst case, the swarm system takes more than N steps, however it is stopped in $M = N$ steps without the stabilization. The worst case leads to a complexity of $O(N^2 + W^2)$. It is important to emphasize that the worst case rarely occurs, requiring a specific configuration of the texture image and even a random image does not produce this special case. In order to analyze the average case, we plot in Fig. 6 the average number of steps needed to converge over 400 images. We can see that the two rules of movement *min* and *max* present similar behavior. Also, the number of agents does not influence the number of steps to converge (e.g. the difference of the average number of steps for $N = 5k$ and $N = 40k$ is only 1.03 steps). Given that $M \sim 13$ for $N = 50k$, the average case leads to a complexity which is very close to the best case, $O(N + W^2)$, and it is a good complexity in comparison to the complexities of Gabor filters $O(W^2 \log W)$ and co-occurrence matrices $O(W^2)$.

5. Experimental results

In order to evaluate the proposed method, experiments were carried out on image datasets of high variability. We first describe such datasets, the experiments to evaluate the parameters of our proposed method, and then the comparative results with the state-of-the-art methods.

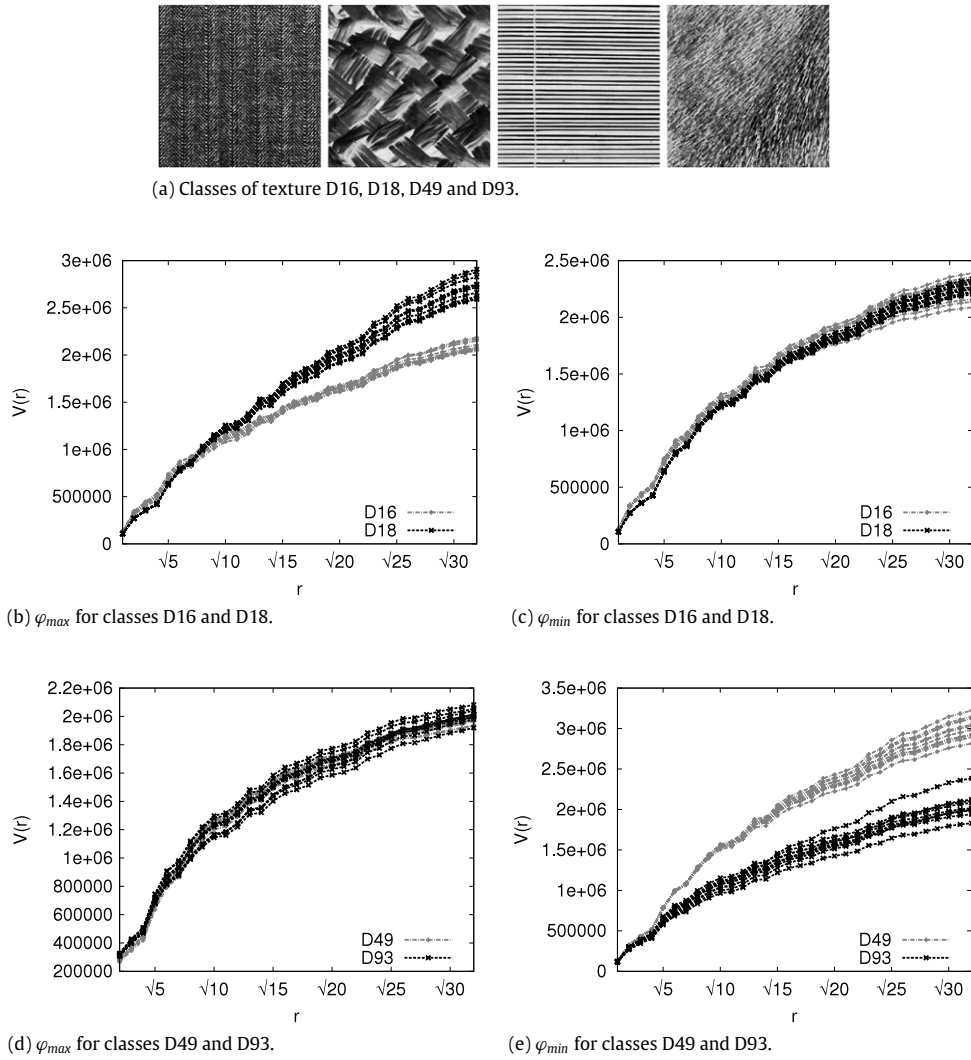


Fig. 5. An example of feature vectors using the rules of movement *min* and *max*. The classes of texture are only discriminated if both rules are used.

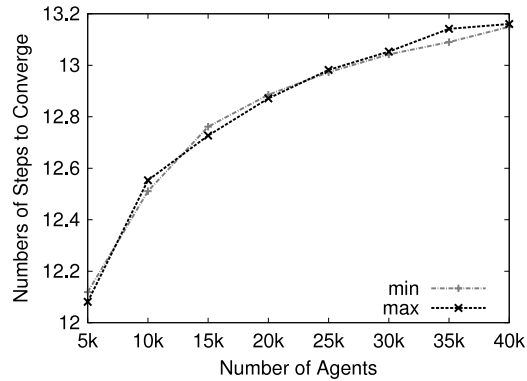


Fig. 6. Average number of steps to converge using the rules of movement *min* and *max*. The average number of steps was averaged over 400 images.

We performed experiments on the two most used image datasets of texture: Brodatz and Vistex. The Brodatz album [45] is the most well known benchmark for evaluating texture methods. Each class is composed by one image divided into ten

non-overlapped samples. The samples have 200×200 pixels with 256 gray levels. In this work, a total of 40 classes with 10 samples per class were used.

Vision Texture — Vistex [46] provides real-world textures under challenging conditions (e.g. lighting and perspective). A total of 54 classes are available, each class containing 16 samples. The samples have 128×128 pixels with 256 gray levels.

In our experiments, linear discriminant analysis (LDA) [47,48] in a 10-fold cross-validation strategy was adopted in the task of classification. The LDA method estimates a linear subspace in which the projection of the vectors presents larger inter-class than intra-class variance. The 10-fold cross-validation strategy divides randomly the samples into 10 folds. Each fold is used to test the classifier while the other nine folds are used to train the classifier. This process is repeated 10 times with each fold used once as testing data. To produce a single statistic, the results of the 10 processes are averaged.

The features used in this paper, and the parameter evaluation are in the next section.

5.1. Parameter evaluation

In this section, we evaluate the three main parameters of our method: number of artificial crawlers N , maximum energy e_{max} and maximum radius r_m of the fractal dimension. The other parameters were set according to Ref. [27], since their possible values do not affect the final success rate. Each artificial crawler is born with initial energy $e_{init} = 10$, the survival threshold $e_{min} = 1$ and the absorption rate $\lambda = 0.01$.

Since the three parameters are dependent, to set a specific configuration we first vary them together to find out the best parameter setting. In this case, we vary the energy from 5k to 40k agents, with the maximum energy (e_{max}) ranging from 5 to 35 and, by using radiuses from $\sqrt{5}$ to $\sqrt{40}$. To reach this goal, experiments were performed by using nested loops varying the three parameters. Thus, we can determine which values we must use in order to obtain the best classification rate. For the Brodatz dataset, the best parameters are $N = 30k$, $e_{max} = 15$, $r_m = \sqrt{37}$, while for the Vistex dataset the best ones are $N = 15k$, $e_{max} = 20$ and $r_m = \sqrt{38}$. Notice that the settings for both datasets are close to each other. For other datasets, we recommend using a number of agents N between 60% and 95% of the number of pixels, $10 \leq e_{max} \leq 25$ and $\sqrt{30} \leq r_m \leq \sqrt{40}$.

In Fig. 7 we present the behavior for each parameter on texture classification. When we want to evaluate the number of agents, the maximum energy and radius parameters are set according to the setting findings aforementioned. The success rates for the different numbers of artificial crawlers are shown in Fig. 7(a) and (b) for Brodatz and Vistex datasets, respectively. The number of artificial crawlers placed on the pixels was initially set to 5k with a coverage rate of 5%, varying from 5k to 40k for the Brodatz dataset and varying from 5k to 15k for the Vistex dataset due to the size of the samples (128×128 pixels). We can observe that the highest success rate was obtained for $N = 30k$ and $N = 15k$ for Brodatz and Vistex, respectively. Further, it was found that the combination of rules *min* and *max* significantly improves the success rate for all numbers of artificial crawlers in both datasets. Also, the rule of going to the minimum intensity provides similar results to the original rule — *max*. These results suggest that the valleys and peaks are important to obtain a robust texture analysis.

The maximum energy of the artificial crawlers is evaluated in the plot of Fig. 7. Fig. 7(c) presents the results for the Brodatz dataset while Fig. 7(d) shows the results for the Vistex dataset. The maximum energy parameter was evaluated by the fact that it limits the artificial crawler energy and, consequently, can limit the fractal dimension space. However, the experimental results show that different values of maximum energy do not influence the success rate considerably. The highest success rate was obtained for $e_{max} = 15$ using the Brodatz dataset and for $e_{max} = 20$ using the Vistex dataset. It can be noted that the same behavior for the rules of movement was obtained here, with the combination of rules providing the highest success rates.

In the plots of Fig. 7(e) and (f), the maximum radius of the fractal dimension estimation is evaluated. As expected, the success rate increases as the radius increases and stabilizes after a certain radius. The maximum radius $r_m = \sqrt{37}$ provided the highest success rate of 99.25% for the Brodatz dataset. For the Vistex dataset, a success rate of 95.95% was obtained by the maximum radius $r_m = \sqrt{38}$. As the previous results, the combination of rules of movement provides the highest success rates. Also, the rule *min* provides similar results compared to the rule *max*.

5.2. Comparison with other methods

The proposed method, which is enriched by the fractal dimension estimation of artificial crawlers, is compared to traditional texture methods, namely Fourier descriptors [7], co-occurrence matrices [49,1], Gabor filter [50,51,8], local binary pattern [52], and multi-fractal spectrum [53]. Moreover, the texture method using the artificial crawlers proposed in Ref. [27] was also used in this comparison. We considered the traditional implementation of each method and its parameter configuration as described below, which yields the best result.

Fourier descriptors: these descriptors are obtained from the Fourier transform of the texture image. Each descriptor is the sum of the spectrum values within a radius from the center. The best results were obtained by radius with increment by one. Thus, for an image of 200×200 pixels, 99 descriptors are obtained. More information about the Fourier descriptors can be found in Ref. [7].

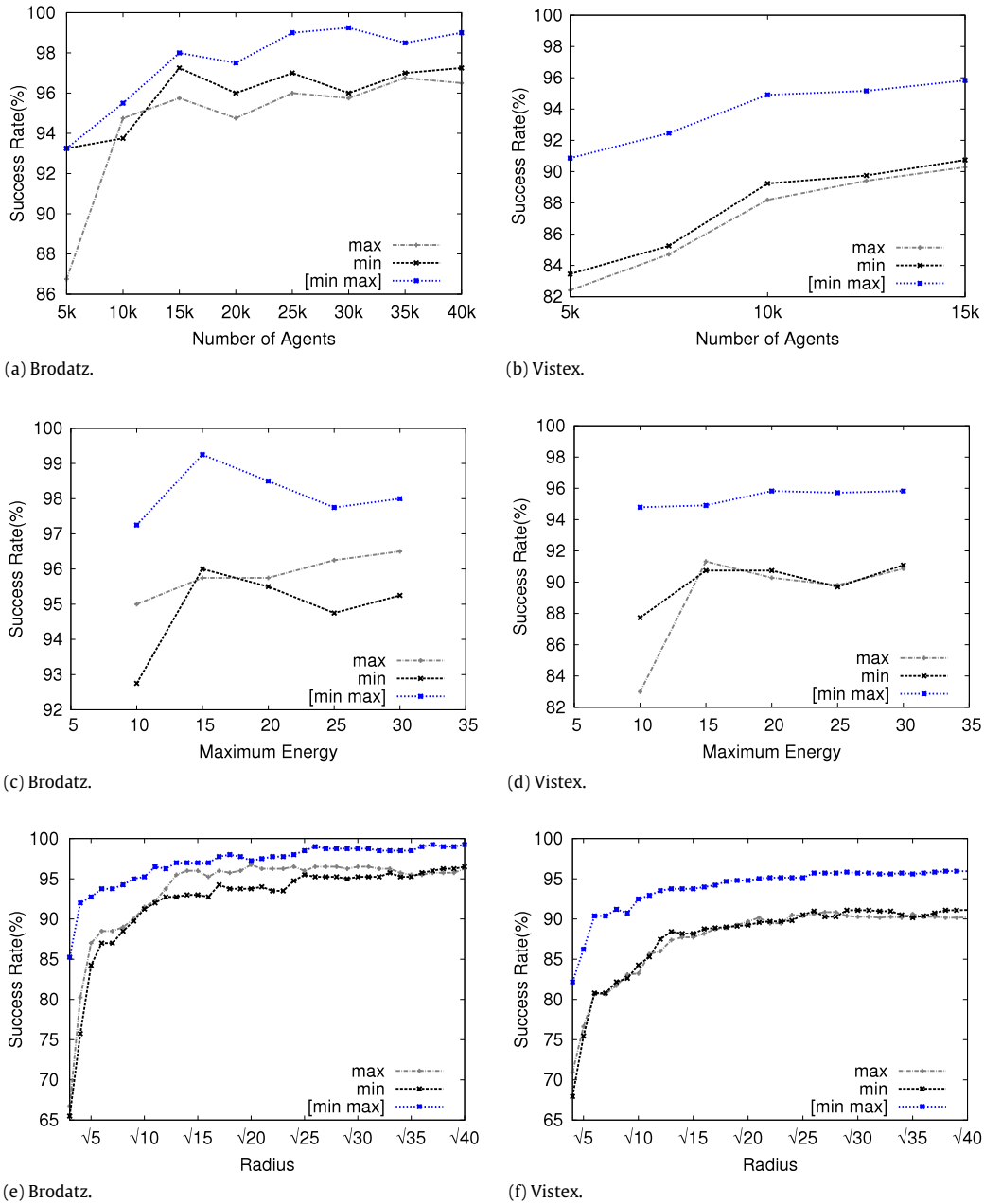


Fig. 7. Plots for different The plot for evaluating the number of artificial crawlers in the Brodatz and Vistex datasets.

Co-occurrence matrices: they are computed by the joint probability distribution between pairs of pixels at a given distance and direction. In these experiments, we consider the distances from 1 to 5 pixels, and the angles 0° , 45° , 90° and 135° . Energy and entropy were calculated from these matrices to compose a 40-dimensional feature vector [1,49].

Gabor filters: it convolves an image by a bank of Gabor filters (i.e., different scales and orientations). In the experiments, a bank of 40 filters (eight rotations and five scales) was used. The energy of each convolved image is used to compose the feature vector; in this case a 40-dimensional feature vector. Additional information can be found in Refs. [50,51,8].

Artificial crawlers: N artificial crawlers, like those explained earlier, are performed in a texture image. Four feature vectors are then calculated: (i) the number of live artificial crawlers at each iteration, (ii) the number of settled artificial crawlers at each iteration, (iii) a histogram of the colony size formed by a certain radius and (iv) scale distribution of the colonies. Finally, the four feature vectors are concatenated to compose a single vector. A complete description of the original method can be found in Refs. [26,27].

Table 1

The experimental results for texture methods in the Brodatz database.

Method	Images correctly classified	Success rate (%)
Fourier descriptors	346	86.50 (± 6.58)
Artificial crawlers	359	89.75 (± 4.76)
Co-occurrence matrices	365	91.25 (± 2.65)
Multi-fractal spectrum	373	93.25 (± 2.37)
Gabor filter	381	95.25 (± 3.43)
Deterministic tourist walk	382	95.50 (± 3.12)
Fractal-DW	398	99.50 (± 1.05)
Local binary patterns	399	99.75 (± 0.79)
Proposed method	397	99.25 (± 1.69)

Table 2

Experimental results for texture methods in the Vistex database.

Method	Images correctly classified	Success rate (%)
Fourier descriptors	672	77.78 (± 4.67)
Artificial crawlers	691	79.98 (± 4.65)
Co-occurrence matrices	663	76.74 (± 4.91)
Deterministic tourist walk	734	84.95 (± 4.13)
Multi-fractal spectrum	747	86.46 (± 3.48)
Gabor filter	774	89.58 (± 2.61)
Fractal-DW	744	86.11 (± 4.42)
Local binary patterns	801	92.71 (± 2.43)
Proposed method	829	95.95 (± 2.50)

Deterministic tourist walk: this method [22] is an agent-based method that builds a joint probability distribution of transient and attractor sizes for different values of memory sizes and two walking rules. In the experiments below, we used memory sizes ranging from 0 to 5.

Fractal-DW: this method can be described into three main steps: (i) attractors are found by deterministic partially self-avoiding walks; (ii) fractal dimension of the attractors is estimated; (iii) feature vector is built based on the dilation process of the fractal dimension estimation. We have followed the parameters suggested in Ref. [23].

Multi-fractal spectrum: this method [53] extracts the fractal dimension of three categorization of the image: intensity, energy of edges, and energy of the Laplacian. For each categorization, a 26-dimensional MFS vector of uniformly spaced values was computed, totaling a feature vector of 78 dimensions.

Uniform rotation-invariant local binary pattern: the LBP method [52] calculates the co-occurrence of gray levels in circular neighborhoods. We used three different spatial resolutions P and three different angular resolutions R — (P, R) : (8, 1), (16, 2) and (24, 3).

In Table 1 we present the comparison of the texture methods on the Brodatz dataset. The proposed method provided comparable results to the local binary patterns and fractal-DW, and superior results to the other state-of-the-art methods. Though local binary patterns features perform slightly better than ours, the experiment also indicates that the proposed method significantly improves the success rate over the original artificial crawlers, i.e., from 89.75% to 99.25%.

Although the Brodatz dataset is widely used for texture classification, it does not contains textures with changes in terms of lighting conditions and perspectives. To evaluate the methods in textures closer to real-world applications, we also compared the results for the Vistex dataset which are presented in Table 2. In this experiment, our method provided the highest success rate of 95.95%, which is superior to the result of the local binary patterns. Our method also significantly improved the success rate compared to the original artificial crawlers. Besides, it can be noted that our method achieved reliable results according to the low standard deviations in both datasets.

6. Conclusion

In this paper we have proposed a new method based on artificial crawlers and fractal dimension for texture classification. We have demonstrated how the feature vector extraction task can be improved by combining two rules of movement, instead of moving only for the maximum intensity of the neighbor pixels. Moreover, a strategy using fractal dimension was proposed to characterize the path of movement performed by the artificial crawlers. The idea of our approach improves the ability of discrimination obtained from the swarm system of artificial crawlers.

Although traditional methods of texture analysis — e.g. Gabor filters, local binary patterns, and co-occurrence matrices — have provided satisfactory results, the method proposed here has proved to be superior for characterizing textures on the Vistex dataset. On the Brodatz album, our method achieve the third place, being slightly inferior to the local binary pattern and fractal-DW methods. Experiments on both datasets indicate that our method significantly improved the classification rate with regard to the original artificial crawler method. As future work, we believe that performance gains can be achieved by means of effective descriptors, for example for representing shape.

Acknowledgments

W.N.G. acknowledges support from FAPESP (# 2010/08614-0). B.B.M. is grateful to FAPESP (# 2011/02918-0). O.M.B. gratefully acknowledges the financial support of CNPq (National Council for Scientific and Technological Development, Brazil) (Grant #308449/2010-0 and #473893/2010-0) and FAPESP (The State of São Paulo Research Foundation) (Grant # 2011/01523-1).

References

- [1] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, *IEEE Trans. Syst. Man Cybern.* 3 (6) (1973) 610–621.
- [2] R.M. Haralick, Statistical and structural approaches to texture, *Proceedings of the IEEE* 67 (5) (1979) 786–804.
- [3] R.L. Kashyap, A. Khotanzad, A model-based method for rotation invariant texture classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1986) 472–481.
- [4] D. Chetverikov, Texture analysis using feature based pairwise interaction maps, *Pattern Recognition* 32 (3) (1999) 487–502.
- [5] G.R. Cross, A.K. Jain, Markov random field texture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5 (1983) 25–39.
- [6] R. Chellappa, S. Chatterjee, Classification of textures using gaussian markov random fields, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33 (1) (1985) 959–963.
- [7] R. Azencott, J.-P. Wang, L. Younes, Texture classification using windowed fourier filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997) 148–153.
- [8] D. Gabor, Theory of communication, *Journal of Institute of Electronic Engineering* 93 (1946) 429–457.
- [9] B.B. Machado, W.N. Gonçalves, O.M. Bruno, Enhancing the texture attribute with partial differential equations: a case of study with gabor filters, in: *Proceedings of the 13th International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS'11*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 337–348.
- [10] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (7) (1989) 674–693. <http://dx.doi.org/10.1109/34.192463>.
- [11] S. Arivazhagan, L. Ganesan, Texture classification using wavelet transform, *Pattern Recognit. Lett.* 24 (9–10) (2003) 1513–1521. [http://dx.doi.org/10.1016/S0167-8655\(02\)00390-2](http://dx.doi.org/10.1016/S0167-8655(02)00390-2).
- [12] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Inc., Orlando, FL, USA, 1983.
- [13] Y. Chen, E. Dougherty, Gray-scale morphological granulometric texture classification, *Optical Engineering* 33 (8) (1994) 2713–2722.
- [14] B.B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Company, New York, 1983.
- [15] O.M. Bruno, R. de Oliveira Plotze, M. Falvo, M. de Castro, Fractal dimension applied to plant identification, *Inform. Sci.* 178 (2008) 2722–2733.
- [16] J. Liu, Y.Y. Tang, Adaptive image segmentation with distributed behavior-based agents, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (6) (1999) 544–551.
- [17] K.-W. Wong, K.-M. Lam, W.-C. Siu, A novel approach for human face detection from color images under complex background, *Pattern Recognition* 34 (10) (2001) 1993–2004.
- [18] V. Rodin, A. Benzinou, A. Guillaud, P. Ballet, F. Harrouet, J. Tisseau, J.L. Bihan, An immune oriented multi-agent system for biological image processing, *Pattern Recognition* 37 (4) (2004) 631–645.
- [19] S.-M. Guo, C.-S. Lee, C.-Y. Hsu, An intelligent image agent based on soft-computing techniques for color image processing, *Expert Systems with Applications* 28 (2005) 483–494.
- [20] J. Jones, M. Saeed, Image enhancement – an emergent pattern formation approach via decentralised multi-agent systems, *Multiagent Grid Systems* 3 (1) (2007) 105–140. URL <http://dl.acm.org/citation.cfm?id=1375348.1375356>.
- [21] S. Mazouzi, Z. Guessoum, F. Michel, A distributed and collective approach for curved object-based range image segmentation, in: *Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, CIARP 2009*, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 201–208.
- [22] A.R. Backes, W.N. Gonçalves, A.S. Martinez, O.M. Bruno, Texture analysis and classification using deterministic tourist walk, *Pattern Recognition* 43 (2010) 685–694.
- [23] W.N. Gonçalves, O.M. Bruno, Combining fractal and deterministic walkers for texture analysis and classification, *Pattern Recognition* 46 (11) (2013) 2953–2968.
- [24] W.N. Gonçalves, O.M. Bruno, Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks, *Expert Systems with Applications* 40 (11) (2013) 4283–4300.
- [25] H. Zheng, A. Wong, S. Nahavandi, Hybrid ant colony algorithm for texture classification, in: *The 2003 Congress on Evolutionary Computation, CEC '03*, Piscataway, N.J., Canberra, Australia, 2003, pp. 2648–2652.
- [26] D. Zhang, Y.Q. Chen, Classifying image texture with artificial crawlers, in: *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT '04*, IEEE Computer Society, Washington, DC, USA, 2004, pp. 446–449.
- [27] D. Zhang, Y.Q. Chen, Artificial life: a new approach to texture classification, *International Journal of Pattern Recognition and Artificial Intelligence* 19 (3) (2005) 355–374.
- [28] C. Tricot, *Curves and Fractal Dimension*, Springer-Verlag, 1995.
- [29] B.B. Mandelbrot, *Fractals: Form, Chance, and Dimension*, in: *Mathematics Series*, W. H. Freeman, San Francisco (CA, USA), 1977.
- [30] F. Hausdorff, Dimension und äusseres mass, *Math. Ann.* 79 (1919) 157–179.
- [31] J. Theiler, Estimating fractal dimension, *Journal of the Optical Society of America A* 7 (6) (1990) 1055–1073.
- [32] D.A. Russell, J.D. Hanson, E. Ott, Dimension of strange attractors, *Phys. Rev. Lett.* 45 (14) (1980) 1175–1178.
- [33] B.B. Chaudhuri, N. Sarkar, Texture segmentation using fractal dimension, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (1995) 72–77.
- [34] S. Peleg, J. Naor, R. Hartley, D. Avnir, Multiple resolution texture analysis and classification, *IEEE Trans. Pattern Anal. Mach. Intell.* 6 (4) (1984) 518–523.
- [35] A. Pentland, Fractal-based description of natural scenes, in: *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 1983, pp. 201–209.
- [36] A. Chaudhuri, C.-C.S. Yan, S.-L. Lee, Multifractal analysis of growing surfaces, *Appl. Surf. Sci.* 238 (1–4) (2004) 513–517.
- [37] D.C. André, R. Backes, O.M. Bruno, Color texture analysis based on fractal descriptors, *Pattern Recognition* 45 (5) (2012) 1984–1992. <http://dx.doi.org/10.1016/j.patcog.2011.11.009>.
- [38] M.d.C. João B. Florindo, André R. Backes, O.M. Bruno, A comparative study on multiscale fractal dimension descriptors, *Pattern Recognition Letters* 33 (6) (2012) 798–806. <http://dx.doi.org/10.1016/j.patrec.2011.12.016>.
- [39] Jao B. Florindo, O.M. Bruno, Fractal descriptors based on fourier spectrum applied to texture analysis, *Physica A* 391 (20) (2012) 4909–4922. <http://dx.doi.org/10.1016/j.physa.2012.03.039>.
- [40] Jao B. Florindo, E.C.P. Mariana, S. Sikora, O.M. Bruno, Characterization of nanostructured material images using fractal descriptors, *Physica A* 392 (7) (2013) 1694–1701. <http://dx.doi.org/10.1016/j.physa.2012.11.020>.
- [41] A.R. Backes, D. Casanova, O.M. Bruno, Plant leaf identification based on volumetric fractal dimension, *Int. J. Pattern Recognit. Artif. Intell.* 23 (6) (2009) 1145–1160.

- [42] R. Fabbri, L.da F. Costa, J.C. Torelli, O.M. Bruno, 2d euclidean distance transform algorithms: a comparative survey, *ACM Comput. Surv.* 40 (1) (2008) 1–44. <http://dx.doi.org/10.1145/1322432.1322434>.
- [43] T. Saito, J.-I. Toriwaki, New algorithms for euclidean distance transformation of an n -dimensional digitized picture with applications, *Pattern Recognition* 27 (11) (1994) 1551–1565. [http://dx.doi.org/10.1016/0031-3203\(94\)90133-3](http://dx.doi.org/10.1016/0031-3203(94)90133-3).
- [44] A. Meijster, J.B.T.M. Roerdink, W.H. Hesselink, A general algorithm for computing distance transforms in linear time, in: *Mathematical Morphology and its Applications to Image and Signal Processing*, 2000, pp. 331–340. http://dx.doi.org/10.1007/0-306-47025-X_36.
- [45] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Publications, New York, 1966.
- [46] S. Singh, M. Sharma, Texture analysis experiments with meastex and vistex benchmarks, in: *Proceedings of the Second International Conference on Advances in Pattern Recognition, ICAPR '01*, Springer-Verlag, London, UK, 2001, pp. 417–424.
- [47] N.H. Timm, *Applied Multivariate Analysis*, in: *Springer Texts in Statistics*, Springer, 2002.
- [48] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed., Academic Press Professional, Inc., San Diego, CA, USA, 1990.
- [49] C. Palm, Color texture classification by integrative co-occurrence matrices, *Pattern Recognition* 37 (5) (2004) 965–976.
- [50] F. Bianconi, A. Fernández, Evaluation of the effects of gabor filter parameters on texture classification, *Pattern Recognition* 40 (12) (2007) 3325–3335.
- [51] A.K. Jain, F. Farokhnia, Unsupervised texture segmentation using gabor filters, *Pattern Recognition* 24 (12) (1991) 1167–1186.
- [52] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7) (2002) 971–987. <http://dx.doi.org/10.1109/TPAMI.2002.1017623>.
- [53] Y. Xu, H. Ji, C. Fermüller, Viewpoint invariant texture description using fractal analysis, *International Journal of Computer Vision* 83 (1) (2009) 85–100. <http://dx.doi.org/10.1007/s11263-009-0220-6>.