Politecnico di Torino

Master of Science in Physics of Complex Systems

Random Textures Analysis via Detrending Moving Average (DMA) Algorithm

Dec 2024

Author:

S. R. Eruppakkattu

Supervisor:

Prof. A. Carbone

Contents

Abstract

Texture classification is an important first step in image segmentation and image recognition. In this report we present the Detrending Moving Average (DMA) algorithm as a robust and informative classification algorithm. We train the DMA algorithm with the images of the UIUC dataset. The Cholensky-Levinson Factorization algorithm is used to generate artificial fractal surfaces as a reference dataset. In the classification results the DMA algorithm seems to be able to detect scale, aspect and rotation changes in the analysed random textures.

I. INTRODUCTION

The world surrounding us is typically characterized by complex structures and patterns that are easily recognized by our eyes. It is indeed *natural* for us to distinguish between a wooden surface from a rock one, or a rough water surface from a plain water surface. The same cannot be said for machines: finding artificial models and algorithms to describe appropriately natural shapes is a challenging task.

From the definition of *fractals*, there has been an increasing research activity in the processing of images as random fractal textures [12–21]. Texture can be defined in fact as a coarseness or roughness measure of a surface and its analysis becomes critical when it allows to distinguish among similar surfaces. There can be various applications as object recognition and image segmentation. Among the various applications we can cite for example the analysis of medical images to assess the health of patients' organs [5], the analysis of food surfaces to check its quality and/or its properties [6, 7], fabric inspection [8] and the characterization of satellite images [9]. If we approach the problem of surface characterization by analysing the underlying "*fractality*" of the surface, the classification problem is reduced to estimating the fractal dimension [16]. This type of investigation seems to be quite robust and informative since the fractal feature is a property inherent to the object itself. Fractals offer then an alternative to classification approaches based on feature vectors, which are not reliable in cases when the lighting conditions are varying [16].

In this framework, we use the Detrending Moving Average (DMA) Algorithm to estimate the fractal dimension D_F *via* the Hurst's exponent H, a parameter that quantifies the scaling

properties of random curves and surfaces [1–4].

The present work will be structured as follows: first we will briefly describe the nature of fractal textures; after, we will present the DMA algorithm, in the one-dimensional and two-dimensional case; then, the DMA algorithm will be used to analyse an artificial dataset of images generated by the Cholensky-Levinson algorithm and the UIUC dataset [21–24]. In the final section we will comment the results obtained and illustrate some future developments and applications.

A. Fractal textures

In the case of random textures exhibiting power law scaling forms, if M is some measure of the surface, such as area or length, when a measuring unit of size λ is used, then

$$M = n\lambda^{D_F} \tag{1}$$

where D_F stands for the fractal dimension, a measure of roughness of the surface. As a direct consequence, smoother surfaces exhibits lower D_F. If we are treating self-similar time series or surfaces then the Hurst exponent H is directly related to the fractal dimension D_F by

$$D_F = d - H \tag{2}$$

where d is the metric dimension of the set [12]. The Hurst exponent H is an empirical parameter [10, 11] which finds its theoretical roots in the framework of the fractional Brownian walk [13]. A one-dimensional fractional Brownian motion (FBM) $B_H(t,\omega)$ is defined by the following equation

$$B_H(t,\omega) = B_H(0,\omega) + \frac{1}{\Gamma(H+\frac{1}{2})}\int_0^t (t-s)^{H-\frac{1}{2}} dB(s) \tag{3}$$

for $t > 0$. The FBM is by construction a moving average of the ordinary Brownian walk in which past increments are weighted by the kernel $(t-s)^{H-\frac{1}{2}}$. Notably, the DMA algorithm relies too on the construction of moving averages. Generalization of the FBM to higher dimensions is straightforward. FBM shows self-similarity of parameter H, where H is the Hurst exponent ranging from 0 to 1, introduced in Eq. (2). For long-memory correlated processes the value of the Hurst exponent H ranges respectively from $0 < H < 0.5$ and from

$0.5 < H < 1$, for negative and positive persistence; $H = 0.5$ corresponds to the ordinary Brownian motion, *i.e.* fully uncorrelated signals.

It is important to note that real images and surfaces are not ideal fractals as defined by the high-dimensional Eq. (3), since the latter are defined to exist at all scales. Real surfaces are characterized by their finite dimension that fixes an upper limit of applicable scales. Thus we cannot expect real images to behave as ideal fractals over all scales. We shall say that a surface is *fractal* if its fractal dimension D_F can be accurately approximated by a scaling law over at least three decades of values [14, 16]. In case of small images, it is of course not possible to detect power law scaling over a wide range of values but still it is possible to extract some relevant information of the surface.

II. METHODS

As mentioned before, the scaling properties of random surfaces can be quantified in terms of the Hurst exponent H, defined in the framework of the fractional Brownian walks. Fractional Brownian functions are characterized by a correlation function depending as a power law on λ [12], where λ defines a measuring unit size, as already used in Eq. (1). The power-law correlation of fractional Brownian functions $f(\boldsymbol{r}) : \mathbb{R}^d \to \mathbb{R}$ can be expressed by the power-law dependence of the variance σ_H,

$$\sigma_H = \left\langle [f(\boldsymbol{r} + \boldsymbol{\lambda}) - f(\boldsymbol{r}]^2 \right\rangle \propto \| \boldsymbol{\lambda} \|^{2H} \tag{4}$$

with $\boldsymbol{r} = (x_1, x_2, ..., x_d)$, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_d)$ and $\| \lambda \| = \sqrt{\lambda_1^2 + \lambda_2^2 + ... + \lambda_d^2}$.

The DMA algorithm is based on a generalized high-dimensional variance of the Brownian function around a moving average; such variance is introduced as follows:

$$\sigma^2_{DMA} = \frac{1}{\mathcal{N}} \sum_{i_1=n_1}^{N_1} \sum_{i_2=n_2}^{N_2} \cdots \sum_{i_d=n_d}^{N_d} [f(i_1, i_2, ..., i_d) - \tilde{f}_{n_1,n_2,...,n_d}(i_1, i_2, ..., i_d)]^2 \tag{5}$$

where $f(i_1, i_2, ..., i_d) = f(\boldsymbol{i})$ is a fractional Brownian function defined over a discrete d-dimensional domain, with maximum sizes $N_1, N_2, ..., N_d$. It is $i_1 = 1, 2, ..., N_1$, $i_2 = 1, 2, ..., N_2$,...,$i_d = 1, 2, ..., N_d$. $\boldsymbol{n} = (n_1, n_2, ..., n_d)$ defines the sub-arrays ν_d of the fractal domain with maximum values $n_{1_{max}} = max(n_1)$, $n_{2_{max}} = max(n_2)$, ..., $n_{d_{max}} = max(n_d)$. The function $\tilde{f}_{n_1,n_2,...,n_d}(i_1, i_2, ..., i_d) = \tilde{f}$ is given by

$$\tilde{f}_{n_1,n_2,...,n_d}(i_1,i_2,...,i_d) = \frac{1}{n_1 n_2 \dots n_d} \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} \cdots \times \sum_{k_d=0}^{n_d-1} f(i_1-k_1, i_2-k_2, ..., i_d-k_d) \quad (6)$$

which is an average of $f(\boldsymbol{i})$ computed over the sub-arrays ν_d. Equations (5) and (6) are defined for any value of $n_1, n_2, ..., n_d$ and for any shape of the sub-arrays, however it is preferable to choose sub-arrays with $n_1 = n_2 = ... = n_d$ to avoid spurious effects.

It was shown in [1] that the generalized variance obeys the following scaling relation:

$$\sigma^2_{DMA} \sim \left(\sqrt{n_1^2 + n_2^2 + \cdots + n_d^2} \right)^{2H} \quad (7)$$

as a consequence of the property (4) of the fractional Brownian functions. In order to calculate the Hurst exponent, the algorithm is implemented through the following steps. The moving average $\tilde{f}$ is computed for different sub-arrays, by varying the sliding windows $n_1, n_2, ..., n_d$ from 2 to the maximum values $n_{1_{max}}, n_{2_{max}}, ..., n_{d_{max}}$. The values $n_{1_{max}}, n_{2_{max}}, ..., n_{d_{max}}$ depend on the size of the fractal domain. In order to minimize the saturation effects due to the finite size of the surface it is better to take $n_{1_{max}} \ll N_1; n_{2_{max}} \ll N_2; ...; n_{d_{max}} \ll N_d$. It is important to note that the basic action of the moving average is that of a low-pass filter, *i.e.* it smooths the fractional Brownian function onto which it acts. Next we will focus on the one-dimensional and two-dimensional case of the DMA algorithm.

A. One-dimensional case

By imposing $d = 1$ in Eq. (4) one obtains

$$\sigma^2_{DMA} = \frac{1}{N_1 - n_{1_{max}}} \sum_{i_1=n_1}^{N_1} [f(i_1) - \tilde{f}(i_1)]^2 \quad (8)$$

where N_1 is the length of the time series and n_1 is the sliding window and $n_{1_{max}} \ll N_1$. Equation (8) defines a a generalized variance of the time series $f(i_1)$ with respect with the function $\tilde{f}(i_1)$ in the following way:

$$\tilde{f}_{n_1}(i_1) = \frac{1}{n_1} \sum_{k_1=0}^{n_1-1} f(i_1 - k_1) \quad (9)$$

which is the moving average of $f(i_1)$, over each sliding window of length n_1. The moving average $\tilde{f}_{n_1}$ is calculated for different values of the window n_1, ranging from 2 to the maximum value $n_{1_{max}}$. The variance σ^2_{DMA} is then calculated according to Eq. (8) and plotted as a function of n_1 on log-log axes. The plot is a straight line, as expected for a power-law dependence of σ^2_{DMA} on n_1 as follows:

$$\sigma^2_{DMA} \sim n_1^{2H} \tag{10}$$

Equation (10) allows one to estimate the scaling exponent H of the series $f(i_1)$. The slope is found through an ordinary least squares (OLS) regression. By the relation (2) for $d = 1$, also the critical dimension D_F can be estimated.

B. Two-dimensional case

For $d = 2$, the generalized variance defined by Eq. (4) writes

$$\sigma^2_{DMA} = \frac{1}{(N_1 - n_{1_{max}})(N_2 - n_{2_{max}})} \times \sum_{i_1=n_1}^{N_1} \sum_{i_2=n_2}^{N_2} [f(i_1, i_2) - \tilde{f}(i_1, i_2)]^2 \tag{11}$$

with $\tilde{f}(i_1, i_2)$ given by

$$\tilde{f}_{n_1}(i_1, i_2) = \frac{1}{n_1 n_2} \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} f(i_1 - k_1, i_2 - k_2) \tag{12}$$

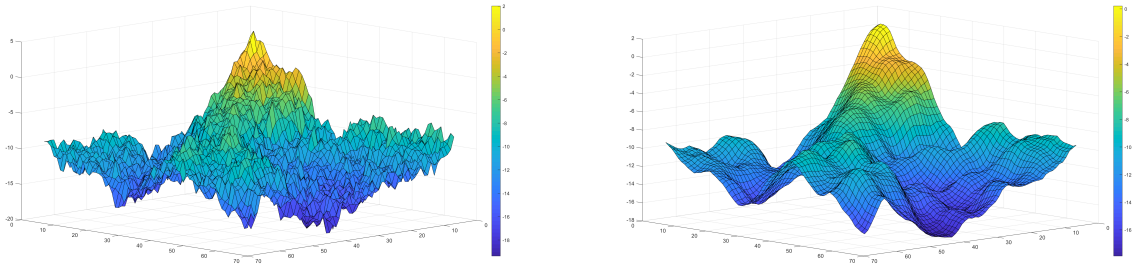
The average $\tilde{f}$ is calculated over sub-arrays with different size $n_1 \times n_2$. The next step is the calculation of the difference $f(i_1, i_2) - \tilde{f}_{n_1,n_2}(i_1, i_2)$ for each sub-array $n_1 \times n_2$. A log-log plot of σ^2_{DMA},

$$\sigma^2_{DMA} \sim \left[\sqrt{n_1^2 + n_2^2}\right]^{2H} = s^{2H} \tag{13}$$

$$\sigma^2_{DMA} \sim \left[\sqrt{n_1^2 + n_2^2}\right]^{2H} = \left[\sqrt{n^2 + n^2}\right]^{2H} = s^{2H} \tag{14}$$

as a function of $s = n_1^2 + n_2^2$, yields a straight line with slope H, found through a OLS regression. Fractal dimension can again be estimated thanks to Eq. (2) by taking $d = 3$. In Fig. 1 one can see the effect of the low-pass filter action on a fractal surface of Hurst exponent $H = 0.5$.

In the present work, the two-dimensional DMA algorithm was developed in MATLAB language.



(a) (b)

FIG. 1: $N_1 \times N_2 = 68 \times 68$ fractal surface (a) and its moving average (b) of sliding window $n = 5$. Through the volume intersections between the fractal surface and its moving mean one can exploit statistical cluster. The fractal surface was generated by the Cholensky-Levinson factorization (CLF) [26].

1. Deviations at the extreme scales

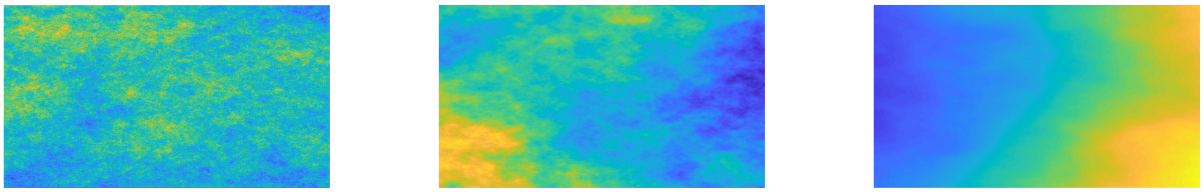
Real images have to deal with finite dimension size, which do not represent the ideal scaling behaviour of theoretical fractals. In case of small surfaces, it is not even possible to prove robust scaling laws over three decades of values as the onset of finite size effects comes into play. At most one can expect to retrieve a local fractal behaviour. This limit is particurarly notable for large scales, as we can see for the log-log plots of the UIUC database images in Fig. 9, Fig. 10, Fig. 11, Fig. 12. The deviations from the linearity at large scales lead to the saturation of the σ^2_{DMA} values. Finite size effects become negligible when the conditions $n_{1_{max}} \ll N_1; n_{2_{max}} \ll N_2; ...; n_{d_{max}} \ll N_d$ are satisfied. Due to the non-ideality of the moving mean at low scales we had to neglect the first data points generated by the DMA algorithm in our simulations.

At small scales the deviations are related to the departure of the moving average from the the ideality. A thorough explanation of this behaviour was given in [2].

III. DATASETS AND RESULTS

We will take the following route: first we will test our method's feasibility and robustness by analysing synthetic rough surfaces with assigned Hurst exponent (Sec. III A), then we will proceed to analyse a collection of images of the UIUC database [21–24] (Sec. III B).

A. Artificial dataset



(a) $H = 0.1$ (b) $H = 0.5$ (c) $H = 0.9$

FIG. 2: Synthetic fractal surfaces generated by the CLF algorithm.

In order to synthetize rough surfaces we used the Cholensky-Levinson factorization algorithm, included in the package FRACLAB [26]. In Fig. 1(a) one can see a synthethized fractal surface of domain size $N_1 \times N_2 = 128 \times 128$ with Hurst exponent $H = 0.5$. Furthermore, in Fig. 2 we show three examples of digital images of domain size $N_1 \times N_2 = 512 \times 512$ mapped to fractal surfaces with reference to the color intensity, *i.e.* to the level of red, green and blu (RGB). Hurst exponents estimated by the proposed method for the three images are, respectively, $H = 0.1$, $H = 0.5$, $H = 0.9$. In Fig. 2 it can be seen how the Hurst exponent is a direct indicator of the correlation of the image's pixels. For $0 < H < 0.5$ the system is anticorrelated (see Fig. 2(a)), for $H = 0.5$ the system is completely random (see Fig. 2(b)) and for $0.5 < H < 1$ the system is positively correlated (see Fig. 2(c)). In order to test the reliability of our method, we generated a dataset of nine artificial fractal surfaces of domain size $N_1 \times N_2 = 490 \times 480$ with the CLF algorithm, with Hurst exponent H ranging from 0.1 to 0.9 with 0.1 increment. This artificial dataset was analysed *via* the two-dimensional DMA algorithm (see Sec. II B). In Fig. 3 are shown the resulting slope estimates (i.e. the Hurst exponents H for each fractal surface) for the artificial dataset. For each Hurst exponent H_{in} given, in Table I are reported the estimation values of the DMA algorithm and the errors associated.

B. UIUC dataset

Among the many texture databases available in the literature, we analysed the University of Illinois Urbana Champaign (UIUC) database [22–24] since its textures exhibited self-

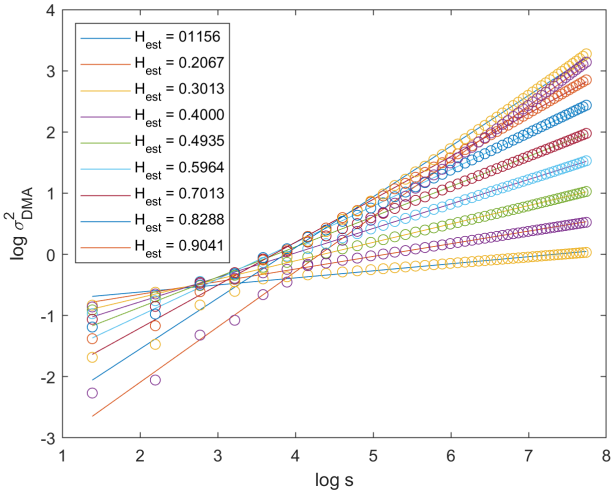


FIG. 3: Log-log plot of σ^2_{DMA} as a function of s for the nine rough surfaces of the artificial dataset of domain size $N_1 \times N_2 = 480 \times 480$.

H_{in}	H_{est}	ΔH_{est}
0.1	0.1156	$+1.56 \times 10^{-2}$
0.2	0.2067	$+6.7 \times 10^{-3}$
0.3	0.3013	$+1.3 \times 10^{-3}$
0.4	0.4000	$+0.0000$
0.5	0.4935	-6.5×10^{-3}
0.6	0.5964	-3.6×10^{-3}
0.7	0.7013	$+1.3 \times 10^{-3}$
0.8	0.8288	$+2.88 \times 10^{-2}$
0.9	0.9041	$+4.1 \times 10^{-3}$

TABLE I: For each artificial fractal surface it is indicated its Hurst exponent H_{in}, its relative estimation through the DMA algorithm and the resulting error ΔH_{est} in the estimation.

similar behaviour in previous studies [21]. The UIUC database contains 1000 images of resolution 640×480 pixels each. The images are organized in 25 different texture classes (T01 - T25) which contain 40 images each. The materials depicted are imaged under significant viewpoint variations and some also have considerable surface deformations (see Fig. 5, 6, 7, 8 for examples). However, a drawback of the dataset is that it has very few instances of a given

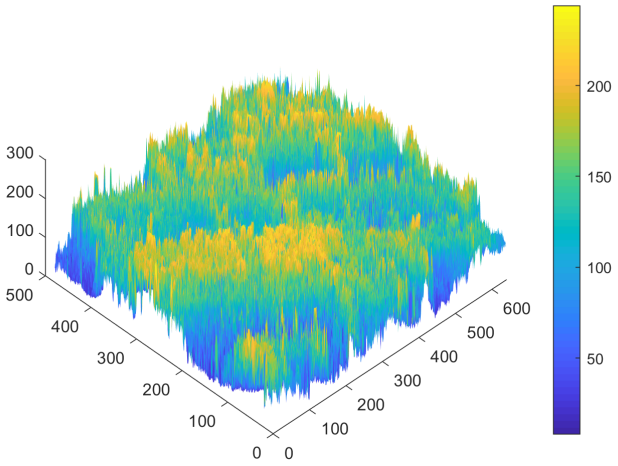


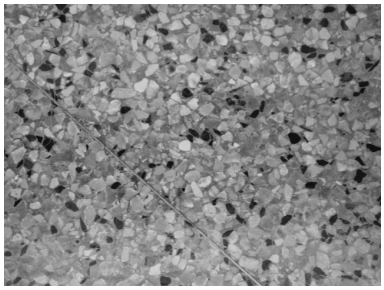
FIG. 4: T14-15 fractal surface in terms of RGB content. The original image can be seen in Fig. 6(c).

material so that it is difficult to perform categorisation experiments or deduce generalisation properties of features [19, 20]. Nevertheless, as far as scale and other viewpoint variations are concerned, the UIUC database is fairly challenging and therefore we test the proposed method on it. In Fig. 4 we can see how the fractal surfaces of the UIUC dataset can be graphically exploited by mapping their color intensity in terms of RGB content.

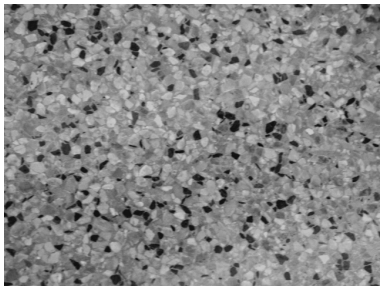
As for the synthetic fractals, before implementing the DMA algorithm we had to choose the upper sliding windows $n_{1_{max}}, n_{2_{max}}$ such that the deviations at large scale were minimized. Since the lower size dimension of the UIUC database images is 480 pixels, we set $n_{1_{max}} = n_{2_{max}} = 48$. By doing so we constructed square sliding windows $n \times n$ increasing with n, from the trivial case $s = 1$ to $s = 48$, whre $s = n \times n$.

C. Results on the UIUC dataset

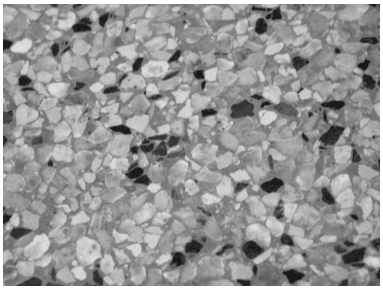
We show in Figs. 5, 6, 7, 8 the log-log plots of σ^2_{DMA} as a function of s for the set of UIUC database textures analysed through the DMA algorithm. In order to better understand the behaviour of the fractals surfaces treated, for each class of the UIUC dataset we compared the slopes of the articial rough surfaces generated by the CLF algorithm (see Sec. III A) with the ones computed for the UIUC dataset's images, for similarity in the values of the Hurst exponents H found. The relative plots are shown in Figs. 9, 10, 11, 12.



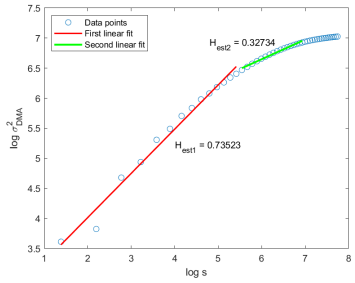
(a) $T11 - 01$



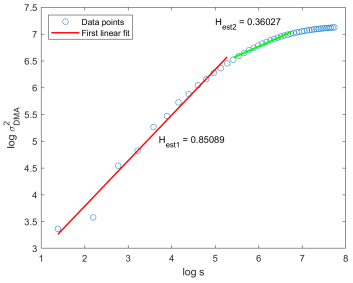
(b) $T11 - 02$



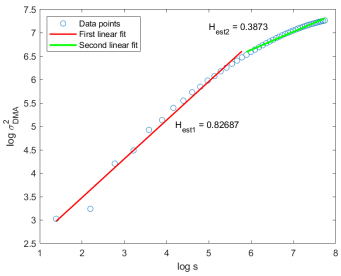
(c) $T11 - 03$



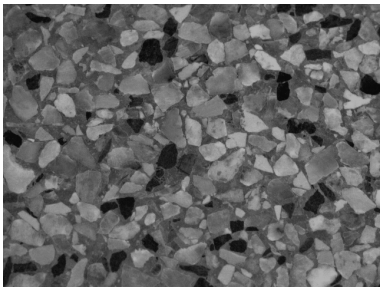
(d) $H_1 \simeq 0.74, H_2 \simeq 0.33$



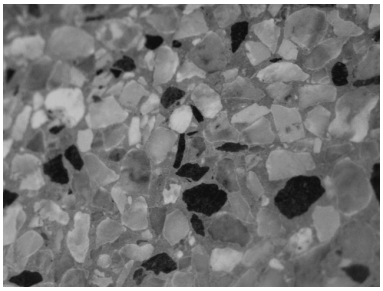
(e) $H_1 \simeq 0.85, H_2 \simeq 0.36$



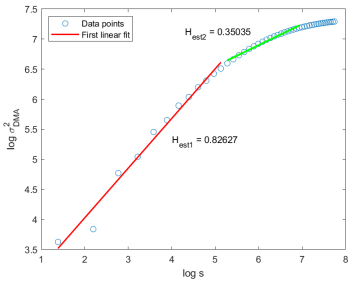
(f) $H_1 \simeq 0.83, H_2 \simeq 0.39$



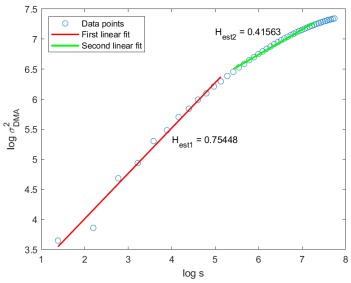
(g) $T11 - 08$



(h) $T11 - 10$

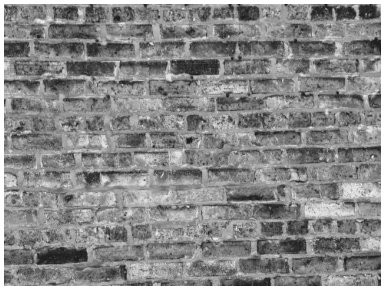


(i) $H_1 \simeq 0.83, H_2 \simeq 0.35$



(j) $H_1 \simeq 0.75, H_2 \simeq 0.42$

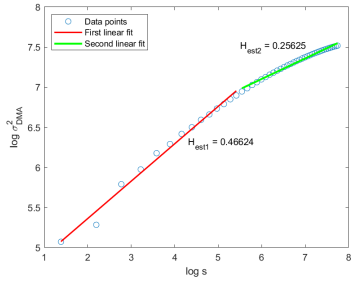
FIG. 5: UIUC textures from the material class T11. Below each image it is shown the log-log plot of σ^2_{DMA} as a function of s. In the log-log plots we can see how the DMA algorithm detects the scaling law of the characteristics size of the pixel clusters, in this case there are two characteristic slopes. The slopes are more evident when the pixel variability is low. The slopes of this class are compared with the closest slopes of the artificial dataset in Fig. 9.



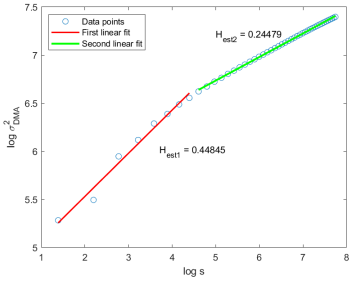
(a) $T14-01$

(b) $T14-06$

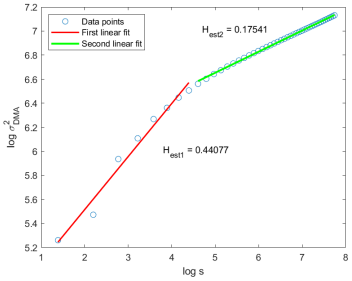
(c) $T14-15$



(d) $H_1 \simeq 0.47, H_2 \simeq 0.26$

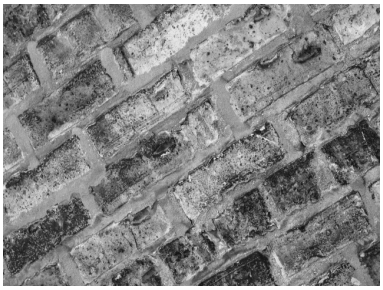


(e) $H_1 \simeq 0.45, H_2 \simeq 0.25$

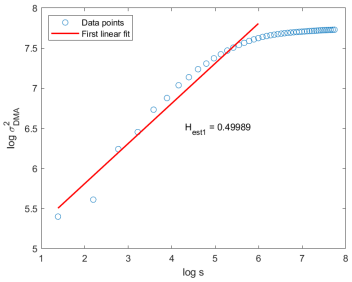


(f) $H_1 \simeq 0.44, H_2 \simeq 0.18$

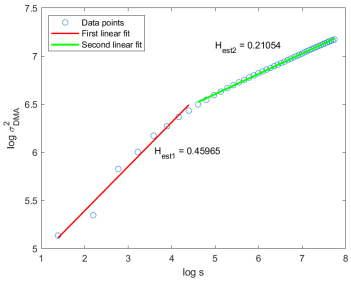
(g) $T14-20$



(h) $T14-38$

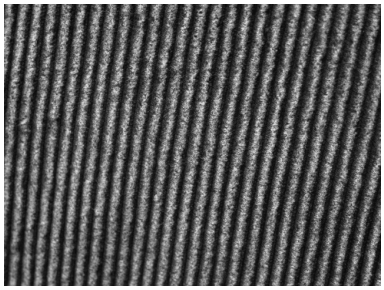


(i) $H_1 \simeq 0.5$

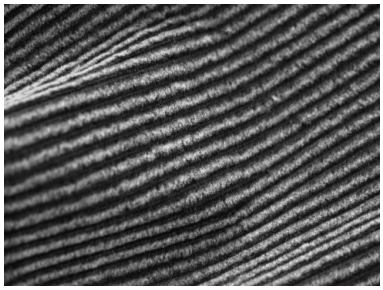


(j) $H_1 \simeq 0.46, H_2 \simeq 0.21$

FIG. 6: UIUC textures from the material class T14. Below each image it is shown the log-log plot of σ^2_{DMA} as a function of s. In the log-log plots we can see how the DMA algorithm detects the scaling law of the characteristics size of the pixel clusters, in this case there are two characteristic slopes. The slopes are more evident when the pixel variability is low. The slopes of this class are compared with the closest slopes of the artificial dataset in Fig. 10.

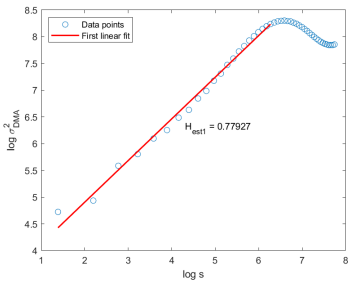


(a) $T24-01$

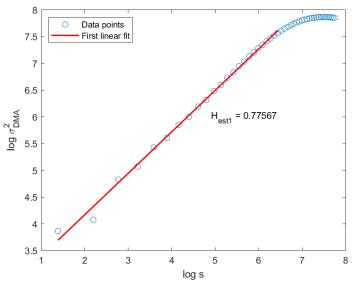


(b) $T24-06$

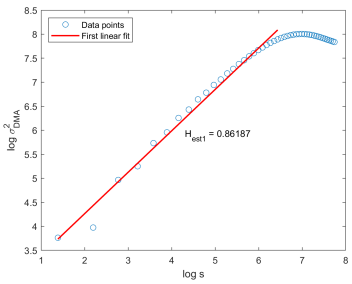
(c) $T24-15$



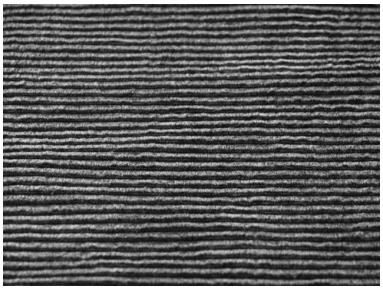
(d) $H \simeq 0.78$



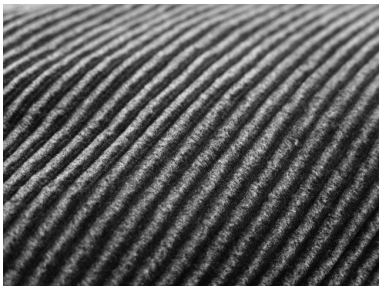
(e) $H \simeq 0.78$



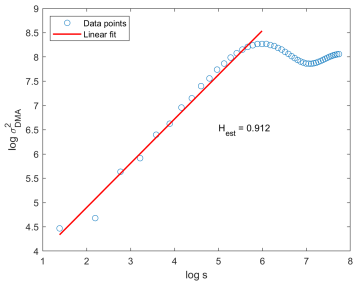
(f) $H \simeq 0.86$



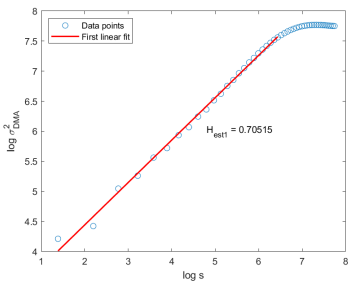
(g) $T24-16$



(h) $T24-20$

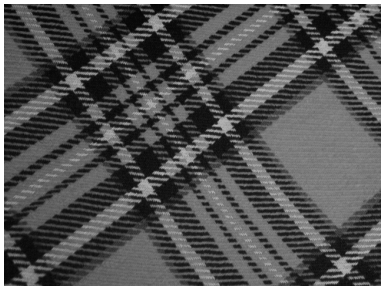


(i) $H \simeq 0.91$

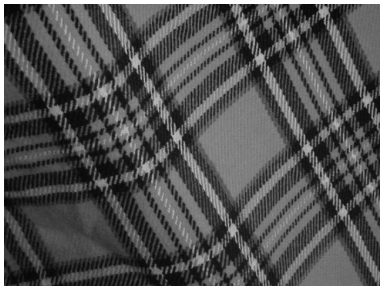


(j) $H = 0.71$

FIG. 7: UIUC textures from the material class T24. Below each image it is shown the log-log plot of σ^2_{DMA} as a function of s. In the log-log plots we can see how the DMA algorithm detects the scaling law of the characteristics size of the pixel clusters, in this case there is one characteristic slope. The slopes are more evident when the pixel variability is low. The slopes of this class are compared with the closest slopes of the artificial dataset in Fig. 11.

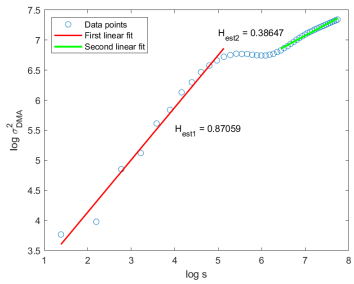


(a) $T25-011$

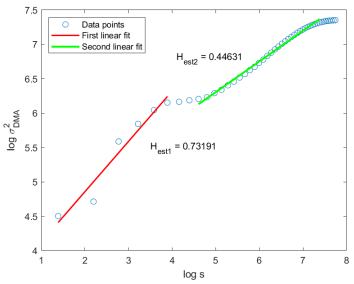


(b) $T25-08$

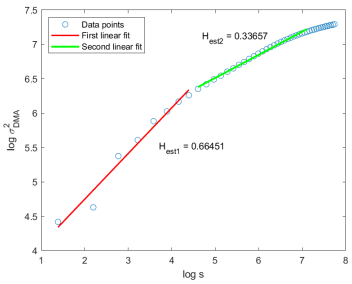
(c) $T25-18$



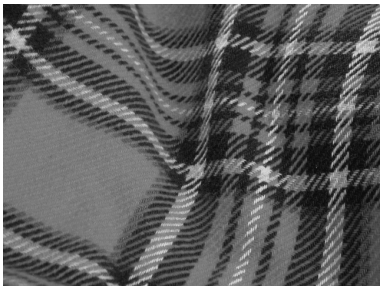
(d) $H_1 \simeq 0.87, H_2 \simeq 0.39$



(e) $H_1 \simeq 0.73, H_2 \simeq 0.45$

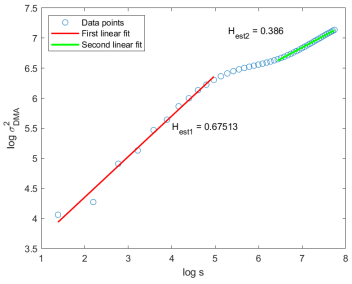


(f) $H_1 \simeq 0.66, H_2 \simeq 0.34$

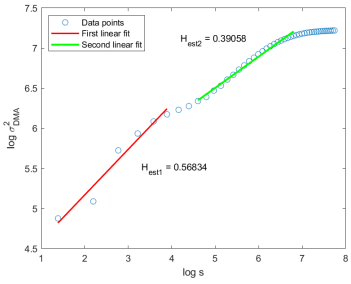


(g) $T25-19$

(h) $T25-32$



(i) $H_1 \simeq 0.67, H_2 \simeq 0.39$



(j) $H_1 \simeq 0.57, H_2 \simeq 0.39$

FIG. 8: UIUC textures from the material class T25. Below each image it is shown the log-log plot of σ^2_{DMA} as a function of s. In the log-log plots we can see how the DMA algorithm detects the scaling law of the characteristics size of the pixel clusters, in this case there are two characteristic slopes. The slopes are more evident when the pixel variability is low. The slopes of this class are compared with the closest slopes of the artificial dataset in Fig. 12.

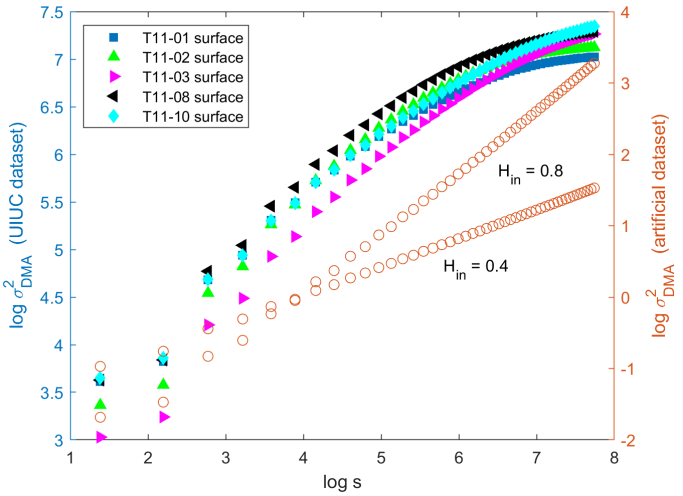


FIG. 9: Comparision between the σ^2_{DMA} log-log plots of the T11 textures (see Fig. 5) and the σ^2_{DMA} log-log plots from the reference artificial fractal surfaces generated by the CLF algorithm (see Fig. 3) with Hurst exponent $H_1 = 0.4$ and $H_2 = 0.8$. The plot has two y-axes, one for the normalized artificial surfaces (right y-axis) and one for the UIUC dataset images (left y-axis).

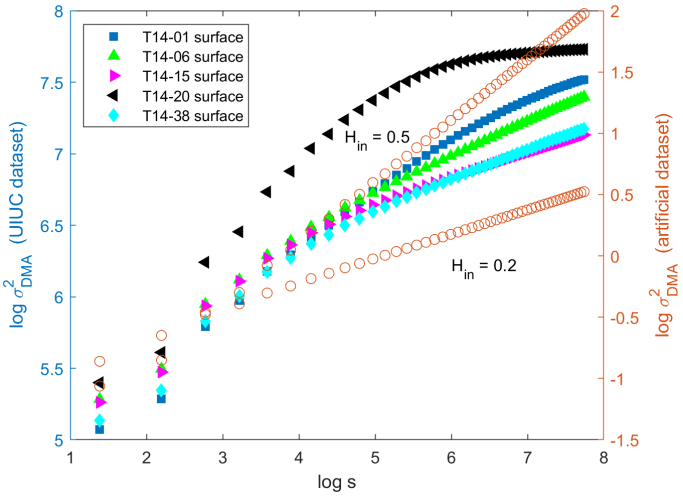


FIG. 10: Comparision between the σ^2_{DMA} log-log plots of the T14 textures (see Fig. 6) and the σ^2_{DMA} log-log plots from the reference artificial fractal surfaces generated by the CLF algorithm (see Fig. 3) with Hurst exponent $H_1 = 0.2$ and $H_2 = 0.5$. The plot has two y-axes, one for the normalized artificial surfaces (right y-axis) and one for the UIUC dataset images (left y-axis).

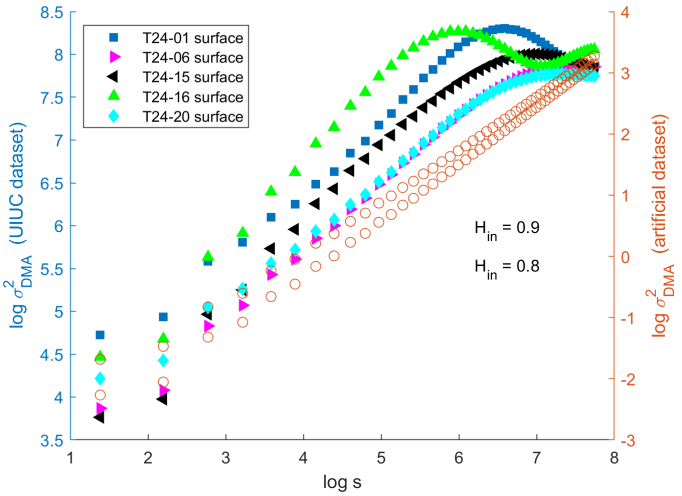


FIG. 11: Comparision between the σ^2_{DMA} log-log plots of the T24 textures (see Fig. 7) and the σ^2_{DMA} log-log plots from the reference artificial fractal surfaces generated by the CLF algorithm (see Fig. 3) with Hurst exponent $H_1 = 0.8$ and $H_2 = 0.9$. The plot has two y-axes, one for the normalized artificial surfaces (right y-axis) and one for the UIUC dataset images (left y-axis).

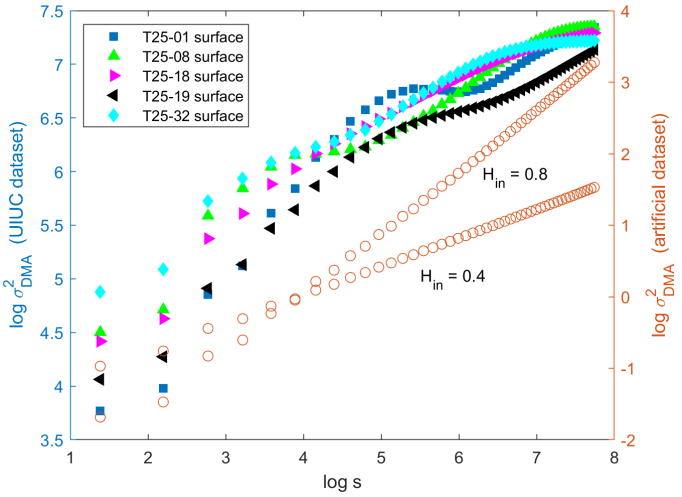


FIG. 12: Comparision between the σ^2_{DMA} log-log plots of the T25 textures (see Fig. 8) and the σ^2_{DMA} log-log plots from the reference artificial fractal surfaces generated by the CLF algorithm (see Fig. 3) with Hurst exponent $H_1 = 0.4$ and $H_2 = 0.8$. The plot has two y-axes, one for the normalized artificial surfaces (right y-axis) and one for the UIUC dataset images (left y-axis).

IV. DISCUSSION AND CONCLUSION

From the results reported in Sec. III C, the DMA algorithm seems to be able to detect the underlying structures of the input UIUC textures, detecting scale, aspect and rotation changes. From the log-log plots of σ^2_{DMA} as a function of s (see Figs. 5, 6, 7, 8) it is evident that the method proposed captures the different structures of the random textures. In particular, when the pixel variability is low, the method proposed is able to detect quite precisely the different shapes present in the texture analysed, as for the case of the different color and size in the bricks of Fig. 6 and the different sizes of the squares in the patchwork fabric of Fig. 8. If the pixel variability increases or if the texture itself does not have structures and shapes over multiple scales (as in Fig. 7), then the DMA algorithm exhibits a saturation or even a drop off (see Fig. 7). As for the Hurst exponent values, these resulted to be higher for the images that presented more evident pixel clusters (*i.e.*, higher pixel-pixel correlation) and low for the images with no clear pixel clusters (*i.e.*, lower pixel-pixel correlation). From the comparison with the fractal surfaces generated with the CLF algorithm (see Figs. 9, 10, 11, 12), the σ^2_{DMA} values of the UIUC dataset textures typically saturate well before the synthetic rough surfaces, as it was expected since the UIUC surfaces have not the ideal properties of true mathematical fractals, due to their finite domain size and lack of structures over all scales.

In view of the results obtained, the present work has shown that the DMA algorithm is a robust and informative method to inquire the properties of random textures, and that it is able to recognize the typical behaviour of a class texture also in different conditions of lighting, rotations and deformations within certain limits. The applications can be various, and the method could be used for image classification and segmentation.

Future developments could include the statistical analysis of clusters formed by the intersections between the fractal surface and its moving average function as done for the one-dimensional case [25], an entropy-based analysis of the rough surfaces as for the one-dimensional case and the extension of the present work to other texture databases.

Acknowledgements

The author would like to thank Prof. A. Carbone for her expertise and advice.

References

[1] E. Alessio, A. Carbone, G. Castelli, V. Frappietro, (2002). "Second-order moving average and scaling of stochastic time series". *Eur. J. Phys. B 27 197.*

[2] A. Carbone (2007), "Algorithm to estimate the Hurst exponent of high-dimensional fractals". *Phys. Rev. E 76, 056703.*

[3] C. Türk, A. Carbone, B.M. Chiaia (2010), "Fractal heterogeneous media". *Phys. Rev. E 81, 026706.*

[4] A. Carbone, B. Chiaia, B. Frigo, C. Trk (2010), "Fractal Model for Snow". *Materials Science Forum, Vols. 638-642, pp. 2555-2560.*

[5] C.-C. Chen, J. S. Daponte, M. D. Fox (1989), "Fractal Feature Analysis and Classification in Medical Imaging", *IEEE Transactions on Medical Imaging, Vol. 8, No. 2.*

[6] F. H. Fan, Q. Ma, J. Ge, Q. Y. Peng, W. W. Riley, S. Z. Tang (2013), "Prediction of texture characteristics from extrusion food surface images using a computer vision system and artificial neural networks". *Journal of Food Engineering, 118(4), 42643.*

[7] R. Quevedo, L.-G. Carlos, J. M. Aguilera, L. Cadoche (2002), "Description of food surfaces and microstructural changes using fractal image texture analysis", *Journal of Food Engineering 361-371.*

[8] A. Conci, C. B. Proença (1998), "A fractal image analysis system for fabric inspection based on a box-counting method", *Computer Networks and ISDN Systems 1887-1895.*

[9] J. C. Valdiviezo-N., R. Castro , G. Cristbal , A. Carbone (2014), "Hurst exponent for fractal characterization of LANDSAT images". *SPIE Optical Engineering + Applications.*

[10] H. E. Hurst (1951), "Long-Term Storage Capacity of Reservoire". *Trans. Amer. Soc. Civ. Engrs, Vol. l16, p. 770.*

[11] H. E. Hurst (1957), "A Suggested Statistical Model of some Time Series which occur in Nature", *Nature 180, 494.*

[12] B.B. Mandelbrot, (1985) "Self-Affine Fractals and Fractal Dimension", *Physica Scripta. Vol. 32, 257-260.*

[13] B. B. Mandelbrot, J. W. Van Ness, (1968), "Fractional Brownian motions, fractional noises and applications." *SIAM Rev. 4, 422.*

[14] A. P. Pentland (1984), "Fractal-based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-6, pp. 661674.*

[15] R. Kashyap, K. Eom (1989), "Texture boundary detection based on the long correlation model, *IEEE Trans. Pattern Anal. Machine Intell., Vol. 11, pp. 5867.*

[16] H. Potlapalli, R. C. Luo (1998), "Fractal-based classification of natural textures" *IEEE Transactions on Industrial Electronics, Vol. 45 , Issue 1.*

[17] M. K. Biswas, T. Ghose, S. Guha, P. K. Biswas (1998), "Fractal dimension estimation for texture images: A parallel approach", *Pattern Recognition Letters 19, 309-313.*

[18] L. M. Kaplan (1999), "Extended Fractal Analysis for Texture Classification and Segmentation", *IEEE Trans. on Image Processing, Vol. 8, No. 11.*

[19] K. J. Dana, B. van Ginneken, S. K. Nayar, J. J. Koenderink (1999), "Reflectance and texture of real world surfaces". *ACM Transactions on Graphics, 18(1):134.*

[20] B. Caputo,E. Hayman, M. Fritz, J.-O. Eklundh (2010), "Classifying materials in the real world". *Image and Vision Computing 28, 150163.*

[21] M. Varma, R. Garg (2007), "Locally Invariant Fractal Features for Statistical Texture Classification". *2007 IEEE 11th International Conference on Computer Vision.*

[22] S. Lazebnik, C. Schmid, and J. Ponce (2005), "A Sparse Texture Representation Using Local Affine Regions", *IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 27 , Iss. 8.*

[23] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid (2007), "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study", *Int. J. Comput. Vision, Vol. 73 Is. 2, pp. 213238.*

[24] The UIUC database is downloadable as *zip* file at `http://slazebni.cs.illinois.edu/research/uiuc_texture_dataset.zip`

[25] A. Carbone, G. Castelli, H.E. Stanley (2004) "Analysis of clusters formed by the moving average of a long-range correlated time series". *Phys. Rev. E 69, 026105.*

[26] We use the CLF algorithm included in the package FRACLAB, downloadable at `https://project.inria.fr/fraclab/`