Algorithm to estimate the Hurst exponent of high-dimensional fractals

Anna Carbone
Physics Department, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy

(Received 19 June 2007; published 13 November 2007)

We propose an algorithm to estimate the Hurst exponent of high-dimensional fractals, based on a generalized high-dimensional variance around a moving average low-pass filter. As working examples, we consider rough surfaces generated by the random midpoint displacement and by the Cholesky-Levinson factorization algorithms. The surrogate surfaces have Hurst exponents ranging from 0.1 to 0.9 with step 0.1, and different sizes. The computational efficiency and the accuracy of the algorithm are also discussed.

DOI: 10.1103/PhysRevE.76.056703 PACS number(s): 05.10.−a, 05.40.−a, 05.45.Df, 68.35.Ct

I. INTRODUCTION

The scaling properties of random curves and surfaces can be quantified in terms of the Hurst exponent H, a parameter defined in the framework of the fractional Brownian walks introduced in [1]. A fractional Brownian function $f(\mathbf{r}):\mathbb{R}^d \to \mathbb{R}$, is characterized by a variance σ_H^2,

$$\sigma_H^2 = \langle [f(\mathbf{r}+\boldsymbol{\lambda}) - f(\mathbf{r})]^2 \rangle \propto \|\boldsymbol{\lambda}\|^{\alpha} \quad \text{with} \quad \alpha = 2H, \tag{1}$$

with $\mathbf{r}=(x_1,x_2,\ldots,x_d)$, $\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\ldots,\lambda_d)$, and $\|\boldsymbol{\lambda}\| = \sqrt{\lambda_1^2+\lambda_2^2+\cdots+\lambda_d^2}$; a power spectrum S_H,

$$S_H \propto \|\boldsymbol{\omega}\|^{-\beta} \quad \text{with} \quad \beta = d + 2H, \tag{2}$$

with $\boldsymbol{\omega}=(\omega_1,\omega_2,\ldots,\omega_d)$ the angular frequency, $\|\boldsymbol{\omega}\| = \sqrt{\omega_1^2+\omega_2^2+\cdots+\omega_d^2}$; and a number of objects N_H of characteristic size ϵ needed to cover the fractal,

$$N_H \propto \epsilon^{-D} \quad \text{with} \quad D = d + 1 - H, \tag{3}$$

D being the fractal dimension of $f(\mathbf{r})$. The Hurst exponent ranges from 0 to 1, taking the values H=0.5, H>0.5, and H<0.5, respectively for uncorrelated, correlated, and anticorrelated Brownian functions.

The application of fractal concepts, through the estimate of H, has been proven useful in a variety of fields. For example, in $d=1$, heartbeat intervals of healthy and sick hearts are discriminated on the basis of the value of H [2,3]; the stage of financial market development is related to the correlation degree of return and volatility series [4]; coding and noncoding regions of genomic sequences have different correlation degrees [5]; and climate models are validated by analyzing long-term correlation in atmospheric and oceanographic series [6,7]. In $d \geqslant 2$ fractal measures are used to model and quantify stress induced morphological transformation [8]; isotropic and anisotropic fracture surfaces [9–13]; static friction between materials dominated by hard core interactions [14]; diffusion [15,16] and transport [17,18] in porous and composite materials; mass fractal features in wet or dried gels [19] and in physiological organs (e.g., lung) [20]; hydrophobicity of surfaces with hierarchic structure undergoing natural selection mechanism [21] and solubility of nanoparticles [22]; and digital elevation models [23] and slope fits of planetary surfaces [24].

A number of fractal quantification methods—based on Eqs. (1)–(3) or on variants of these relationships—such as rescaled range analysis (R/S), detrended fluctuation analysis (DFA), detrending moving average analysis (DMA), and spectral analysis, have been thus proposed to accomplish accurate and fast estimates of H in order to investigate correlations at different scales in $d=1$. A comparatively small number of methods able to capture spatial correlations, operating in $d \geqslant 2$, have been proposed so far [25–32]. This work is addressed to develop an algorithm to estimate the Hurst exponent of high-dimensional fractals and thus is intended to capture scaling and correlation properties over space. The proposed method is based on a generalized high-dimensional variance of the fractional Brownian function around a moving average. In Sec. II, we report the relationships holding for fractals with arbitrary dimension. It is argued that the implementation can be carried out in directed or isotropic mode. We show that the detrending moving average (DMA) method [30–32] is recovered for $d=1$. In Sec. III, the feasibility of the technique is proven by implementing the algorithm on rough surfaces—with different size $N_1 \times N_2$ and Hurst exponent H—generated by the random midpoint displacement (RMD) and by the Cholesky-Levinson factorization (CLF) methods [33,34]. The generalized variance is estimated over subarrays $n_1 \times n_2$ with different size ("*scales*") and then averaged over the whole fractal domain $N_1 \times N_2$. This feature reduces the bias effects due to nonstationarity with an overall increase of accuracy—compared to the two-point correlation function, whose average is calculated over all the fractal. Furthermore—compared to the two-point correlation function, whose implementation is carried out along one-dimensional lines (e.g., for the fracture problem, the two-point correlation functions are measured along the crack propagation direction and the perpendicular one), the present technique is carried out over d-dimensional structures (e.g., squares in $d=2$). In Sec. IV, we discuss the accuracy and range of applicability of the method.

II. METHOD

In order to implement the algorithm, the generalized variance σ_{DMA}^2 is introduced as follows:

$$\sigma_{DMA}^2 = \frac{1}{\mathcal{N}} \sum_{i_1=n_1-m_1}^{N_1-m_1} \sum_{i_2=n_2-m_2}^{N_2-m_2} \cdots \sum_{i_d=n_d-m_d}^{N_d-m_d} [f(i_1,i_2,\ldots,i_d) - \widetilde{f}_{n_1,n_2,\ldots,n_d}(i_1,i_2,\ldots,i_d)]^2, \tag{4}$$

where $f(i_1,i_2,\ldots,i_d)=f(\mathbf{i})$ is a fractional Brownian function defined over a discrete d-dimensional domain, with maximum sizes $N_1,N_2,\ldots,N_d$. It is $i_1=1,2,\ldots,N_1$, $i_2=1,2,\ldots,N_2,\ldots$, $i_d=1,2,\ldots,N_d$. $\mathbf{n}=(n_1,n_2,\ldots,n_d)$ defines the subarrays ν_d of the fractal domain with maximum values $n_{1\,\max}=\max\{n_1\}$, $n_{2\,\max}=\max\{n_2\},\ldots$. $n_{d\,\max}=\max\{n_d\}$; $m_1=\mathrm{int}(n_1\theta_1)$, $m_2=\mathrm{int}(n_2\theta_2),\ldots$, $m_d=\mathrm{int}(n_d\theta_d)$ and θ_1, θ_2, $\ldots\theta_d$ are parameters ranging from 0 to 1; $\mathcal{N}=(N_1-n_{1\,\max})(N_2-n_{2\,\max})\cdots(N_d-n_{d\,\max})$. The function $\widetilde{f}_{n_1,n_2,\ldots,n_d}(i_1,i_2,\ldots,i_d)=\widetilde{f}$ is given by

$$\widetilde{f}_{n_1,n_2,\ldots,n_d}(i_1,i_2,\ldots,i_d) = \frac{1}{n_1 n_2\cdots n_d}\sum_{k_1=-m_1}^{n_1-1-m_1}\sum_{k_2=-m_2}^{n_2-1-m_2}\cdots \times\sum_{k_d=-m_d}^{n_d-1-m_d} f(i_1-k_1,i_2-k_2,\ldots,i_d-k_d), \qquad (5)$$

which is an average of f calculated over the subarrays ν_d. Equations (4) and (5) are defined for any value of $n_1,n_2,\ldots,n_d$ and for any shape of the subarrays, however, it is preferable to choose subarrays with $n_1=n_2=\cdots=n_d$ to avoid spurious directionality in the results. The generalized variance σ^2_{DMA} varies as $(\sqrt{n_1^2+n_2^2+\cdots+n_d^2})^{2H}$ as a consequence of the property (1) of the fractional Brownian functions.

Upon variation of the parameters $\theta_1,\theta_2,\ldots,\theta_d$ in the range [0,1], the indexes $i_1,i_2,\ldots,i_d$ and $k_1,k_2,\ldots,k_d$ of the sums in Eqs. (4) and (5) are accordingly set within ν_d. In particular, $(i_1,i_2,\ldots,i_d)$ coincides, respectively, with (a) one of the vertices of ν_d for $\theta_1=\theta_2=\cdots=\theta_d=0$ and $\theta_1=\theta_2=\cdots=\theta_d=1$ or (b) the center of ν_d for $\theta_1=\theta_2=\cdots=\theta_d=1/2$. It is worthy of note that the choice $\theta_1=\theta_2=\cdots=\theta_d=1/2$ corresponds to the *isotropic* implementation of the algorithm, while $\theta_1=\theta_2=\cdots=\theta_d=0$ and $\theta_1=\theta_2=\cdots=\theta_d=1$ correspond to the *directed* implementation. For example, in $d=2$, the *isotropic* implementation implies that the variance defined by Eq. (4) is referred to a moving average $\widetilde{f}$ calculated over squares $n_1\times n_2$ whose center is (i_1,i_2). Conversely, the *directed* implementation implies that the function $\widetilde{f}$ is calculated over squares $n_1\times n_2$ with one of the vertices in (i_1,i_2). The *directed* mode is of interest to estimate H in fractals with preferential growth direction, e.g., biological tissues (lung), epitaxial layers, and crack propagation in fracture (*anisotropic fractals*). If the fractal is *isotropic* and the accuracy is a priority, the parameters $\theta_1,\theta_2,\ldots,\theta_d$ should be preferably taken equal to 1/2 to achieve the most precise estimate of H. The dependence of the algorithm on θ for $d=1$ has been discussed in [32].

In order to calculate the Hurst exponent, the algorithm is implemented through the following steps. The moving average $\widetilde{f}$ is calculated for different subarrays ν_d, by varying $n_1,n_2,\ldots,n_d$ from 2 to the maximum values $n_{1\,\max},n_{2\,\max},\ldots,n_{d\,\max}$. The values $n_{1\,\max},n_{2\,\max},\ldots,n_{d\,\max}$ depend on the maximum size of the fractal domain. In order to minimize the saturation effects due to finite size, it should be $n_{1\,\max}\ll N_1$; $n_{2\,\max}\ll N_2;\ldots;n_{d\,\max}\ll N_d$. These constraints will be further clarified in Sec. III, where the algorithm is implemented over fractal surfaces with different sizes. For each subarray ν_d, the corresponding value of σ^2_{DMA} is calculated and finally plotted on log-log axes.

To elucidate the way the algorithm works, in the following we consider its implementation for $d=1$ and $d=2$. The case $d=1$ reduces to the detrending moving average (DMA) method already used for long-range correlated time series [30–32].

One-dimensional case. By posing $d=1$ in Eq. (4), one obtains

$$\sigma^2_{DMA} = \frac{1}{N_1-n_{1\,\max}}\sum_{i_1=n_1-m_1}^{N_1-m_1}[f(i_1)-\widetilde{f}_{n_1}(i_1)]^2, \qquad (6)$$

where N_1 is the length of the sequence, n_1 is the sliding window, and $n_{1\,\max}=\max\{n_1\}\ll N_1$. The quantity $m_1=\mathrm{int}(n_1\theta_1)$ is the integer part of $n_1\theta_1$ and θ_1 is a parameter ranging from 0 to 1. The relationship (6) defines a generalized variance of the sequence $f(i_1)$ with respect to the function $\widetilde{f}_{n_1}(i_1)$ as follows:

$$\widetilde{f}_{n_1}(i_1) = \frac{1}{n_1}\sum_{k_1=-m_1}^{n_1-1-m_1} f(i_1-k_1), \qquad (7)$$

which is the moving average of $f(i_1)$ over each sliding window of length n_1. The moving average $\widetilde{f}_{n_1}(i_1)$ 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. (6) 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:

$$\sigma^2_{DMA} \sim n_1^{2H}. \qquad (8)$$

Equation (8) allows one to estimate the scaling exponent H of the series $f(i_1)$. Upon variation of the parameter θ_1 in the range [0,1], the index k_1 in $\widetilde{f}_{n_1}(i_1)$ is accordingly set within the window n_1. In particular, $\theta_1=0$ corresponds to the average $f_{n_1}(i_1)$ over all the points to the left of i_1 within the window n_1; $\theta_1=1$ corresponds to the average $f_{n_1}(i_1)$ over all the points to the right of i_1 within the window n_1; $\theta_1=\frac{1}{2}$ corresponds to the average $f_{n_1}(i_1)$ with the reference point in the center of the window n_1.

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

$$\sigma^2_{DMA} = \frac{1}{(N_1-n_{1\,\max})(N_2-n_{2\,\max})} \times\sum_{i_1=n_1-m_1}^{N_1-m_1}\sum_{i_2=n_2-m_2}^{N_2-m_2}[f(i_1,i_2)-\widetilde{f}_{n_1,n_2}(i_1,i_2)]^2, \qquad (9)$$

with $\widetilde{f}_{n_1,n_2}(i_1,i_2)$ given by

$$\widetilde{f}_{n_1,n_2}(i_1,i_2) = \frac{1}{n_1 n_2} \sum_{k_1=-m_1}^{n_1-1-m_1} \sum_{k_2=-m_2}^{n_2-1-m_2} f(i_1-k_1, i_2-k_2). \quad (10)$$

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

$$\sigma^2_{DMA} \sim [\sqrt{n_1^2+n_2^2}]^{2H} \sim s^H \quad (11)$$

as a function of $s = n_1^2 + n_2^2$, yields a straight line with slope H.

Depending upon the values of the parameters θ_1 and θ_2, entering the quantities $m_1 = \text{int}(n_1\theta_1)$ and $m_2 = \text{int}(n_2\theta_2)$ in Eqs. (9) and (10), the position of (k_1,k_2) and (i_1,i_2) can be varied within each subarray. (i_1,i_2) coincides with a vertex of the subarray if (i) $\theta_1=0$, $\theta_2=0$; (ii) $\theta_1=0$, $\theta_2=1$; (iii) $\theta_1=1$, $\theta_2=0$; and (iv) $\theta_1=1$, $\theta_2=1$ (*directed implementation*). The choice $\theta_1=\theta_2=1/2$ corresponds to take the point (i_1,i_2) coinciding with the center of each subarray $n_1 \times n_2$ (*isotropic implementation*) [41].

III. RESULTS

In order to test feasibility and robustness of the proposed method, synthetic rough surfaces with assigned Hurst exponents have been generated by the random midpoint displacement (RMD) and by the Cholesky-Levinson factorization (CLF) algorithms [33,34]. The widespread use of the RMD algorithm is based on the trade-off of its fast, simple, and efficient implementation to the limited accuracy, especially for $H \ll 0.5$ and $H \gg 0.5$. Conversely, the Cholesky-Levinson factorization algorithm is one of the most accurate techniques to generate 1D and 2D fractional Brownian functions, at the expense of a more complex implementation structure [42].

In Fig. 1, the log-log plots of σ^2_{DMA} as a function of s are shown for the synthetic fractal surfaces generated by the RMD (circles) and by the CLF method (squares). The surfaces have Hurst exponents H_{in} ranging from 0.1 to 0.9 with step 0.1. The domain sizes are, respectively, $N_1 \times N_2 = 256 \times 256$ (a), $N_1 \times N_2 = 1024 \times 1024$ (b), and $N_1 \times N_2 = 4096 \times 4096$ (c). The dashed lines show the behavior that should be exhibited by variances varying exactly as $s^{H_{in}}$ over the entire range of scales. The plots of σ^2_{DMA} as a function s are in good agreement with the behavior expected on the basis of Eq. (11). The quality of the fits is higher for the surfaces generated by the CLF method, confirming that the RMD algorithm synthesizes less accurate fractals. By comparing the results of the simulation (symbols) to the straight lines corresponding to full linearity over the whole range (dashed), deviations from the full linearity can be observed especially for the small surfaces at the extremes of the scale. A plot of the slopes for the fractal surfaces generated by the CLF algorithm is shown in Fig. 2 for different sizes of the fractal domain. A detailed discussion of the origin of the deviations at low and large scales is reported in the Sec. IV.

Finally, we show three examples of digital images currently mapped to fractal surfaces with reference to the color intensity, i.e., to the level of red, green, and blue (RGB). The Hurst exponents estimated by the proposed method are, respectively, $H=0.1$ (a), $H=0.5$ (b), and $H=0.9$ (c) for the images in Fig. 3.

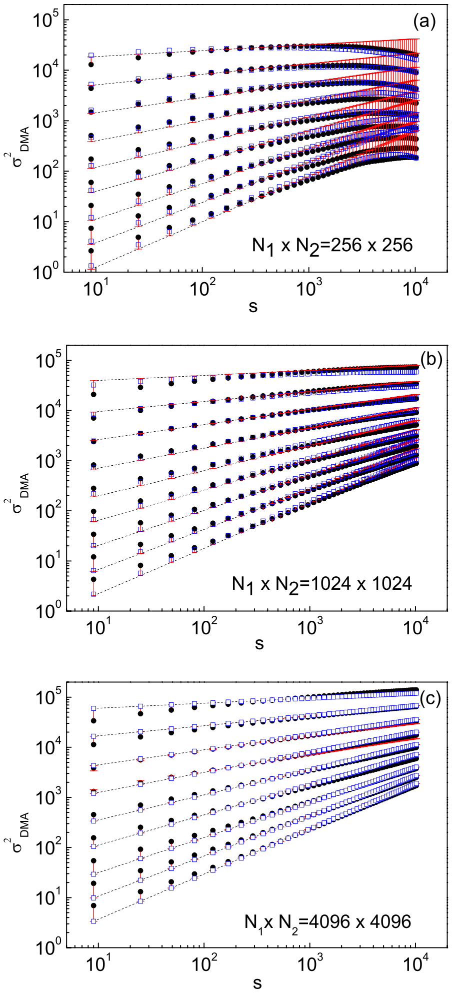

FIG. 1. (Color online) Log-log plot of σ^2_{DMA} for fractal surfaces, respectively, with size $N_1 \times N_2 = 256 \times 256$ (a), $N_1 \times N_2 = 1024 \times 1024$ (b), and $N_1 \times N_2 = 4096 \times 4096$ (c). The data refer to fractal surfaces generated by the RMD (black circles) and by the CLF (blue squares) methods. The Hurst exponent H_{in}—input of the RMD and CLF algorithms—varies from 0.1 to 0.9 with step 0.1. The results correspond to the isotropic implementation, i.e., with the parameters $\theta_1 = \theta_2 = 1/2$ in the Eq. (9). The dashed lines represent the behavior expected for full linearity, i.e., the log-log plot of curves varying as $s^{H_{in}}$. It is worthy of note that the CLF results are much closer to the full-linear behavior compared to the RMD ones.

IV. DISCUSSION

The proposed algorithm is characterized by short execution time and ease of implementation. By considering the case $d=2$, the function $\widetilde{f}_{n_1,n_2}(i_1,i_2)$ is indeed simply obtained by summing the values of $f(i_1,i_2)$ over each subarray $n_1 \times n_2$. Then the sum is updated at each step by adding the last and discarding the first row (column) of each sliding array $n_1 \times n_2$. For higher dimensions, the sum is updated at each

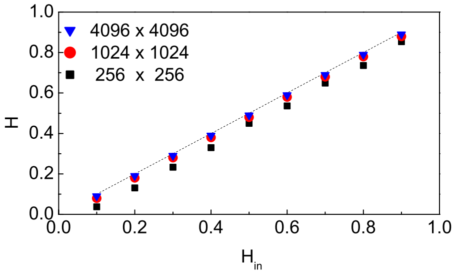

FIG. 2. (Color online) Plot of the values of H obtained by linear fit of the data shown in Figs. 1(a)–1(c) obtained by implementing the proposed algorithm on the fractal surfaces generated by the Cholesky-Levinson factorization method. The dashed line represents the ideal behavior: $H=H_{in}$.

step by adding and discarding a $d-1$-dimensional set of each array $n_1 \times n_2 \times \cdots \times n_d$. The algorithm does not use arbitrary parameters, the computation simply relying on averages of f. We will now argue on the origin of the deviations at small and at large scales.

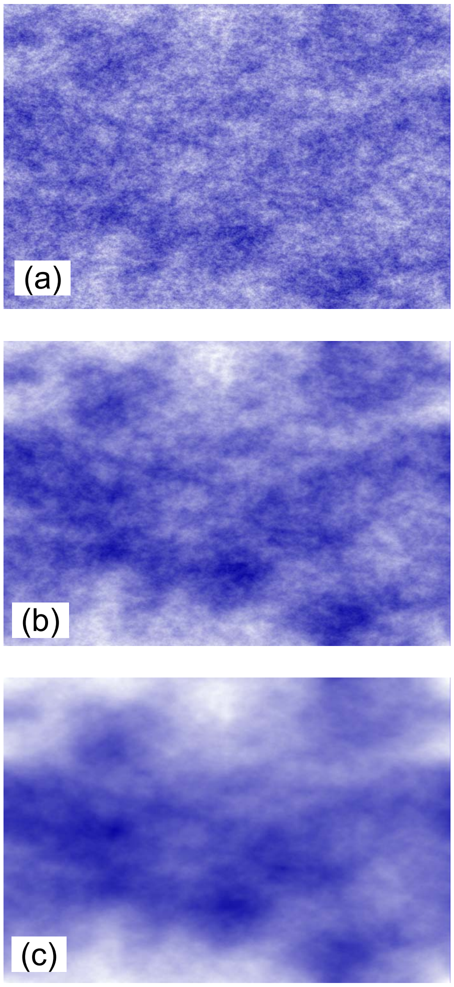

FIG. 3. (Color online). Cloudy sky images, respectively, with Hurst exponent $H=0.1$ (a), $H=0.5$ (b), and $H=0.9$ (c). Such heterogeneous surfaces can be represented as fractals by mapping their color intensity in terms of RGB content.

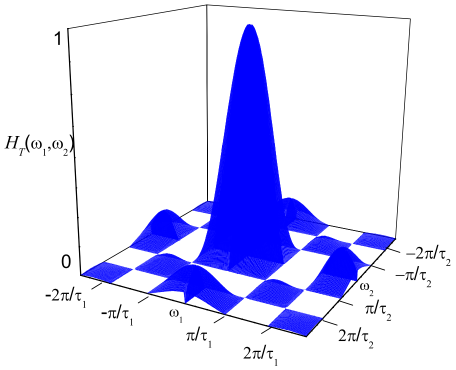

FIG. 4. (Color online) Plot of the transfer function $\mathcal{H}_{\mathcal{T}}(\omega_1,\omega_2)$.

Deviations occurring at large scales. The deviations from the linearity at large scales, leading to the saturation of the σ^2_{DMA} values, are due to finite size effects. The small surfaces do not contain enough data to make the evaluation of the scaling law over the subarrays statistically meaningful. By comparing the data in Figs. 1(a)–1(c), one can note that the saturation effect reduces upon increasing the size $N_1 \times N_2$ of the fractal surface. The finite size effects become negligible when the conditions $n_{1\,\max} \ll N_1$; $n_{2\,\max} \ll N_2; \ldots; n_{d\,\max} \ll N_d$ are fulfilled.

Deviations occurring at small scales. The deviations occurring at low scales are related to the departure of the low-pass filter from the ideality. This problem also occurs with one-dimensional fractals (time series) resulting in the quite generally reported overestimation of H in anticorrelated signals and underestimation of H in correlated signals [35–39]. We will discuss the origin of these deviations by means of the filter transfer function $\mathcal{H}_{\mathcal{T}}(\omega)$ [40]. The algorithm is based on a generalized variance of the function f with respect to $\tilde{f}$. The function $\tilde{f}$ is the output of a *low-pass filter* driven by f, with impulse response a box-car function. In the Appendix, the transfer function $\mathcal{H}_{\mathcal{T}}(\omega)$ of $\tilde{f}$ is explicitly calculated and shown in Fig. 4 for $d=2$. For an *ideal* low-pass filter, the transfer function should be one or zero, respectively, at frequencies lower or higher than the cutoff frequency. However, in real low-pass filters, at frequencies lower than the cutoff frequency, all the components of the signal suffer some attenuation but $\omega=0$. The cutoff frequencies of $\mathcal{H}_{\mathcal{T}}(\omega)$ are $\omega_i=\pi/\tau_i$, i.e., the first zeroes of the functions $\sin \omega_i\tau_i/\omega_i\tau_i$ in the Eq. (A4). Moreover, in real filters, at frequencies higher than π/τ_i, due to the presence of the sidelobes, components of the signals lying in the bands $(\pi/\tau_i, 2\pi/\tau_i); (2\pi/\tau_i, 3\pi/\tau_i); \ldots$, are not fully filtered out. As a result, the function $\tilde{f}$ contains (a) less components with frequency lower than $\omega_i=\pi/\tau_i$ and (b) more components with frequency higher than $\omega_i=\pi/\tau_i$ compared to what would be expected with an ideal low-pass filter. The lack of low-frequency components depends on the central lobe, while the excess of high-frequency components depends on the sidelobes. The excess of high-frequency components results in the decrease of the difference $f-\tilde{f}$, i.e., in the decrease of σ^2_{DMA} and, finally, in the increase of the slope of the log-log plot. Conversely, the lack of low-frequency compo-

TABLE I. Slopes H_I, H_{II}, H_{III}, and relative errors ΔH_I, ΔH_{II}, ΔH_{III} of the curves plotted in Fig. 1(b) (squares) with $N_1 \times N_2 = 1024 \times 1024$. The slopes have been calculated by linear fit, respectively, over the ranges $10 \leqslant s \leqslant 100$ (H_I), $10 \leqslant s \leqslant 1000$ (H_{II}), and $10 \leqslant s \leqslant 10000$ (H_{III}).

H_{in}	H_I	ΔH_I	H_{II}	ΔH_{II}	H_{III}	ΔH_{III}
0.1	0.1346	$+3.46 \times 10^{-1}$	0.1073	$+7.30 \times 10^{-2}$	0.0718	-2.822×10^{-1}
0.2	0.2272	$+1.36 \times 10^{-1}$	0.2050	$+2.50 \times 10^{-2}$	0.1700	-1.500×10^{-1}
0.3	0.3233	$+7.77 \times 10^{-2}$	0.2995	-1.67×10^{-3}	0.2716	-9.467×10^{-2}
0.4	0.4205	$+5.12 \times 10^{-2}$	0.3970	-7.50×10^{-3}	0.3691	-7.725×10^{-2}
0.5	0.5178	$+3.56 \times 10^{-2}$	0.4973	-5.40×10^{-3}	0.4752	-4.960×10^{-2}
0.6	0.6171	$+2.85 \times 10^{-2}$	0.5973	-4.50×10^{-3}	0.5617	-6.383×10^{-2}
0.7	0.7185	$+2.64 \times 10^{-2}$	0.6956	-6.29×10^{-3}	0.6770	-3.286×10^{-2}
0.8	0.8207	$+2.58 \times 10^{-2}$	0.7999	-1.25×10^{-4}	0.7659	-4.263×10^{-2}
0.9	0.9253	$+2.81 \times 10^{-2}$	0.8999	-1.11×10^{-4}	0.8679	-3.567×10^{-2}

nents results in the increase of the difference $f-\tilde{f}$, i.e., in the increase of σ^2_{DMA} and, finally, in the decrease of the slope of the log-log plot. The two effects are more relevant with smaller values of the scales, when the filter nonideality is greater. Moreover, as one can deduce from Eqs. (2) and (A6), the effect of the sidelobes dominates in high-frequency rich fractals with $H<0.5$, while the effect of the central lobe is dominant in low-frequency rich fractals with $H>0.5$.

In order to gain further insight in the above theoretical arguments, we report in Table I the slopes H_I, H_{II}, and H_{III} of the curves (squares) plotted in Fig. 1(b) over different ranges. The slopes have been calculated by linear fit, respectively, over the ranges $10 \leqslant s \leqslant 100$ (H_I), $10 \leqslant s \leqslant 1000$ (H_{II}), and $10 \leqslant s \leqslant 10000$ (H_{III}). The relative errors $\Delta H = (H - H_{in})/H_{in}$ are given, respectively, in the third, fifth, and seventh columns. The slope H_I is greater than the expected value H_{in}. The slope H_{II} is overestimated for $H=0.1$ and $H=0.2$ and underestimated for $H>0.2$. The slope H_{III} is underestimated since the effects of the finite size of the fractal domain play a dominant role.

We address the question if the artifacts due to the filter nonideality described above might be corrected somehow. In the remaining of this section, we will thus consider the use of *windows* whose general effect is to increase the width of the central lobe while reducing those of the sidelobes of the function $\mathcal{H}_{\mathcal{I}}(\omega)$ (a detailed description of these methods can be found in [40]). By restricting our discussion to the present technique, the correction is performed by using the following variant of the relationship (5):

$$\tilde{f}^{\star}_{n_1,n_2,\ldots,n_d}(i_1,i_2,\ldots,i_d) = (1-\alpha) f_{n_1,n_2,\ldots,n_d}(i_1,i_2,\ldots,i_d) + \alpha \tilde{f}_{n_1,n_2,\ldots,n_d}(i_1-1,i_2-1,\ldots,i_d-1), \qquad (12)$$

where $\alpha = n_1 n_2, \ldots, n_d / [(n_1+1)(n_2+1)\cdots(n_d+1)]$. Equation (12) reduces for $d=1$ to the exponentially weighted moving average (EWMA). In practice, the difference between Eq. (5) and Eq. (12) is that the function $\tilde{f}^{\star}$ places more importance to the data around the point $i_1, i_2, \ldots, i_d$. This is achieved by assigning to the function a weight $(1-\alpha)$, while all the other values are summed together and weighted by α. In Fig. 5, we show the ratio $\sigma^2_{DMA}/s^{H_{in}}$ obtained by implementing the algorithm, respectively, with the function $\tilde{f}$ (solid lines) and $\tilde{f}^{\star}$ (dashed lines) in the range $10 \leqslant s \leqslant 100$. To avoid the overlap in the scaled variance obtained for different values of H_{in}, the $\sigma^2_{DMA}/s^{H_{in}}$ has been shifted. The plots are shown in logarithmic scales to allow the correction by direct comparison with the data plotted in log scales in Fig. 1. The ratio $\sigma^2_{DMA}/s^{H_{in}}$ is noticeably closer to a constant when the implementation is performed with $\tilde{f}^{\star}$, with a corresponding reduction of two orders of magnitude in the relative error ΔH_I of the slope H_I.

V. CONCLUSION

We have put forward an algorithm to estimate the Hurst exponent of fractals with arbitrary dimension, based on the high-dimensional generalized variance σ^2_{DMA} defined by Eq. (4).

The methods currently used to estimate the Hurst exponent of high-dimensional fractals are based on: (i) 1-d two-point correlation and structure functions operated along dif-

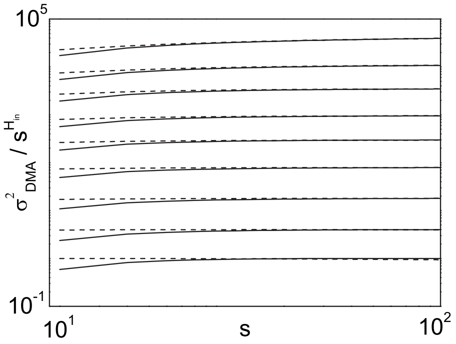

FIG. 5. Plot of the function $\sigma^2_{DMA}/s^{H_{in}}$, respectively, with $\tilde{f}$ defined by the Eq. (4) (solid lines) and $\tilde{f}^{\star}$ defined by the Eq. (12) (dashed lines). The data refer to fractal surfaces generated by the CLF algorithm with $N_1 \times N_2 = 1024 \times 1024$. It can be noted that the deviations from the constant behavior at small scales are reduced with $\tilde{f}^{\star}$ implying a corresponding reduction of the relative error ΔH_I in the slope H_I.

ferent directions, and (ii) high-d Fourier and wavelet transforms [9–13,29]. The advantage of the methods (i) is the ease of implementation. Their drawback is the limited accuracy due to biases and nonstationarities, being these functions are calculated over the entire fractal domain. The methods (ii) are more accurate, however, their implementation is complicated especially for data set with limited extension. The generalized variance σ^2_{DMA} is *scaled*, meaning that it is calculated over subarrays of the whole fractal domain by means of the function $\tilde{f}$. The *scales* are set by the size of the subarrays $n_1 \times n_2 \times \cdots . \times n_d$. Therefore, the proposed method exhibits at the same time (a) ease of implementation, being based on a variancelike approach and (b) high accuracy, being calculated over scaled subarrays rather than on the whole fractal domain.

A further important feature of the proposed algorithm is that it can be implemented "isotropically" or in "directed" mode to accomplish estimates of H in fractals having preferential growth direction, e.g., biological tissues, epitaxial layers, or in crack propagation in fracture. The *isotropic implementation* is obtained by taking $\theta_1=\theta_2=\cdots=\theta_d=1/2$ in Eq. (4). This choice implies that the reference point $(i_1,i_2,\ldots,i_d)$ of the moving average lies in the center of each subarray $n_1\times n_2\times\cdots\times n_d$ and thus $\tilde{f}$ is calculated by summing the values of f around $(i_1,i_2,\ldots,i_d)$. Conversely, to implement the algorithm in a preferential direction (*directed implementation*), the reference point must be coincident with one of the extremes of the segment n_1, or with one of the vertices of the square grid $n_1\times n_2$ or of the d-dimensional array $n_1\times n_2\times\cdots\times n_d$. The directed implementation can be performed by choosing, for example, $\theta_1=\theta_2=\cdots=\theta_d=0$.

Further generalizations of the proposed method can be envisaged for application to the analysis of time-dependent spatial correlations in $d \geqslant 2$.

APPENDIX: TRANSFER FUNCTION OF $\tilde{f}$

The function $\tilde{f}$, defined by Eq. (5), corresponds to the discrete form of the integral as follows:

$$\tilde{f}(x_1,x_2,\ldots,x_d)=\frac{1}{\tau_1\tau_2\cdots\tau_d}\int_{x_1-\tau_1}^{x_1}dx_1' \times\int_{x_2-\tau_2}^{x_2}dx_2'\cdots\int_{x_d-\tau_d}^{x_d}dx_d' f(x_1',x_2',\ldots,x_d'), \tag{A1}$$

where for the sake of simplicity we have considered the case $\theta_1=\theta_2=,\ldots,=\theta_d=0$. Equation (A1) can be rewritten as a convolution integral as follows:

$$\tilde{f}(x_1,x_2,\ldots,x_d)=\frac{1}{\tau_1\tau_2\cdots\tau_d}\int_{-\infty}^{\infty}dx_1^* U\left(\frac{x_1^*}{\tau_1}\right)\int_{-\infty}^{\infty}dx_2^* U\left(\frac{x_2^*}{\tau_2}\right)\cdots \times\int_{-\infty}^{\infty}dx_d^* U\left(\frac{x_d^*}{\tau_d}\right) f(x_1-x_1^*,x_2-x_2^*,\ldots,x_d-x_d^*), \tag{A2}$$

with the convolution kernels given by the boxcar function as follows:

$$U(x_i^*/\tau_i)=\begin{cases}1 & \text{for } 0<x^*/\tau_i<1\\ 0 & \text{elsewhere.}\end{cases}$$

The transfer function can be calculated as follows:

$$\mathcal{H}_{\mathcal{T}}(\omega_1,\omega_2,\ldots,\omega_d)=\frac{1}{\tau_1\tau_2\cdots\tau_d}\int_0^{\tau_1}dx_1\int_0^{\tau_2}dx_2\cdots \times\int_0^{\tau_d}dx_d \exp[-i2\pi(\omega_1x_1+\omega_2x_2+\cdots+\omega_dx_d)] \tag{A3}$$

that can be written as

$$\mathcal{H}_{\mathcal{T}}(\omega_1,\omega_2,\ldots,\omega_d)=\prod_{i=1}^{d}\frac{\sin\omega_i\tau_i}{\omega_i\tau_i} \tag{A4}$$

that is thus d times the function $\sin \omega_i\tau_i/\omega_i\tau_i$.

The Fourier transform $\tilde{\mathcal{F}}$ of the function $\tilde{f}$ can be obtained by means of the following relationship:

$$\tilde{\mathcal{F}}(\omega_1,\omega_2,\ldots,\omega_d)=\mathcal{H}_{\mathcal{T}}(\omega_1,\omega_2,\ldots,\omega_d)\mathcal{F}(\omega_1,\omega_2,\ldots\omega_d), \tag{A5}$$

where $\mathcal{F}(\omega_1,\omega_2,\ldots,\omega_d)$ is the Fourier transform of the function $f(x_1,x_2,\ldots,x_d)$. The power spectrum $\tilde{S}$ of the function $\tilde{f}$ is given by

$$\tilde{S}(\omega_1,\omega_2,\ldots,\omega_d)=|\mathcal{H}_{\mathcal{T}}(\omega_1,\omega_2,\ldots,\omega_d)|^2 S(\omega_1,\omega_2,\ldots,\omega_d), \tag{A6}$$

where $S(\omega_1,\omega_2,\ldots,\omega_d)$ is the power spectrum of the function $f(x_1,x_2,\ldots,x_d)$.

[1] B. B. Mandelbrot and J. W. Van Ness, SIAM Rev. **4**, 422 (1968).

[2] S. Thurner, M. C. Feurstein, and M. C. Teich, Phys. Rev. Lett. **80**, 1544 (1998).

[3] A. L. Goldberger, L. A. N. Amaral, J. M. Hausdorff, P. Ch. Ivanov, C.-K. Peng, and H. E. Stanley, Proc. Natl. Acad. Sci. U.S.A. **99**, 2466 (2002).

[4] T. Di Matteo, T. Aste, and M. M. Dacorogna, J. Banking &amp; Finance **29**, 827 (2005).

[5] C. K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, Phys. Rev. E **49**, 1685 (1994).

[6] Y. Ashkenazy, D. Baker, H. Gildor, and S. Havlin, Geophys. Res. Lett. **30**, 2146 (2003).

[7] P. Huybers and W. Curry, Nature (London) **441**, 7091 (2006).

[8] D. L. Blair and A. Kudrolli, Phys. Rev. Lett. **94**, 166107 (2005).

[9] L. Ponson, D. Bonamy, and E. Bouchaud, Phys. Rev. Lett. **96**, 035506 (2006).

[10] A. Hansen, G. G. Batrouni, T. Ramstad, and J. Schmittbuhl, Phys. Rev. E **75**, 030102(R) (2007).

[11] E. Bouchbinder, I. Procaccia, S. Santucci, and L. Vanel, Phys. Rev. Lett. **96**, 055509 (2006).

[12] J. Schmittbuhl, F. Renard, J. P. Gratier, and R. Toussaint, Phys. Rev. Lett. **93**, 238501 (2004).

[13] S. Santucci, K. J. Maloy, A. Delaplace, J. Mathiesen, A. Hansen, Jan Oistein Haavig, J. Schmittbuhl, L. Vanel, and P. Ray, Phys. Rev. E **75**, 016104 (2007).

[14] J. B. Sokoloff, Phys. Rev. E **73**, 016104 (2006).

[15] P. Levitz, D. S. Grebenkov, M. Zinsmeister, K. M. Kolwankar, and B. Sapoval, Phys. Rev. Lett. **96**, 180601 (2006).

[16] K. Malek and M. O. Coppens, Phys. Rev. Lett. **87**, 125505 (2001).

[17] E. N. Oskoee and M. Sahimi, Phys. Rev. B **74**, 045413 (2006).

[18] M. Filoche and B. Sapoval, Phys. Rev. Lett. **84**, 5776 (2000).

[19] D. R. Vollet, D. A. Donatti, A. Ibanez Ruiz, and F. R. Gatto, Phys. Rev. B **74**, 024208 (2006).

[20] B. Suki, A.-L. Barabasi, Z. Hantos, F. Petak, and H. E. Stanley, Nature (London) **368**, 615 (1994).

[21] R. Garcia, C. J. Gómez, N. F. Martinez, S. Patil, C. Dietz, and R. Magerle, Phys. Rev. Lett. **97**, 016103 (2006).

[22] A. Mihranyan and M. Stromme, Surf. Sci. **601**, 315 (2007).

[23] P. E. Fisher and N. J. Tate, Prog. Phys. Geogr. **30**, 467 (2006).

[24] A. K. Sultan-Salem and G. L. Tyler, J. Geophys. Res., [Planets] **111**, E06S07 (2006).

[25] S. Davies and P. Hall, J. R. Stat. Soc. Ser. B (Stat. Methodol.) **61**, 147 (1999).

[26] G. Rangarajan and M. Ding, Phys. Rev. E **61**, 004991 (2000).

[27] J. Alvarez-Ramirez, J. C. Echeverria, I. Cervantes, and E. Rodriguez, Physica A **361**, 677 (2006).

[28] G. F. Gu and W. X. Zhou, Phys. Rev. E **74**, 061104 (2006).

[29] P. Kestener and A. Arneodo, Phys. Rev. Lett. **91**, 194501 (2003).

[30] E. Alessio, A. Carbone, G. Castelli, and V. Frappietro, Eur. Phys. J. B **27**, 197 (2002).

[31] A. Carbone, G. Castelli, and H. E. Stanley, Phys. Rev. E **69**, 026105 (2004); A. Carbone and H. E. Stanley, Physica A **340**, 544 (2004).

[32] S. Arianos and A. Carbone, Physica A **382**, 9 (2007).

[33] R. H. Voss, in *Random Fractal Forgeries*, edited by R. A. Earnshaw, NATO Advanced Studies Institute series, Vol. F17 Fundamental Algorithm for Computer Graphics (Springer-Verlag, Berlin/Heidelberg, 1985).

[34] W.-X. Zhou and D. Sornette, Int. J. Mod. Phys. C **13**, 137 (2002).

[35] D. C. Caccia, D. Percival, M. J. Cannon, G. Raymond, and J. B. Bassingthwaighte, Physica A **246**, 609 (1997).

[36] K. Hu, P. Ch. Ivanov, Z. Chen, P. Carpena, and H. E. Stanley, Phys. Rev. E **64**, 011114 (2001).

[37] Z. Chen, P. Ch. Ivanov, K. Hu, and H. E. Stanley, Phys. Rev. E **65**, 041107 (2002).

[38] L. M. Xu, P. C. Ivanov, K. Hu, Z. Chen, A. Carbone, and H. E. Stanley, Phys. Rev. E **71**, 051101 (2005).

[39] S. Stoev, M. S. Taqqu, C. Park, G. Michailidis, and J. S. Marron, Comput. Stat. Data Anal. **50**, 2447 (2006).

[40] R. W. Hamming, *Digital Filters* (Prentice-Hall, Englewood Cliffs, NJ, 1998).

[41] The source and executable files of the proposed algorithm can be downloaded at www.polito.it/noiselab/utilities

[42] We use the CLF algorithm included in the package FRACLAB downloadable at http://www.irccyn.ec-nantes.fr/hebergement/FracLab/