



IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

On the phase of natural stochastic structured textures and its application in deconvolution and image decomposition

**Ido Zachevsky and
Yehoshua Y. Zeevi**

CCIT Report #898
May 2016

 Electronics
Computers
Communications

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL



On the phase of natural stochastic structured textures and its
application in deconvolution and image decomposition

Ido Zachevsky and Yehoshua Y. Zeevi

Contents

1	Introduction	2
2	Background	3
2.1	Magnitude and phase of NST	3
2.2	Fractional Brownian motion	3
2.3	Fractal behaviour on discrete and limited supported patches	7
3	Phase analysis for 1D and 2D signals	9
3.1	The phase of one-dimensional signals	9
3.2	The phase of two-dimensional signals	16
4	Reconstruction	22
4.1	Noise	25
4.2	Blur	26
4.3	Deconvolution of anisotropic textures	26
4.4	Example results	28
5	A combined model for isotropic and anisotropic textures	32
5.1	Definition of the model	32
5.2	Estimation of α for a 2D signal	37
5.3	Finding a suitable representation for structured textures	41
5.4	Application in texture decomposition	46
6	Discussion	51
A	The variance of the sample variance of correlated Gaussian variables	55
B	The shifted-phase DCT matrix	57

C Orthogonality of the non-separable DCT transform	59
D Analytical solution to the modified SSF problem	61
E MMSE-based decomposition	63

1 Introduction

The study of textures plays an important role in image understanding, processing and analysis. The decomposition of textured images into their structured skeletons (the so-called cartoon component) and textural components is advantageous in many research areas, including compression, interpolation, classification, characterization, to name a few (e.g. [1, 2, 3]).

Textures are loosely classified into three groups: structured (regular) textures, irregular textures and natural stochastic textures (NST) [4]. Regular textures are comprised of a repeated geometric pattern or a strictly periodic behaviour. NST, on the other hand, are characterised by their statistical properties.

The statistical properties of textures contain information about fine details that are often lost and/or degraded during image acquisition, processing and communication. Algorithms that perform successfully in processing of images containing a limited amount of texture, fail to produce satisfactory results when applied on fully textured images [5]. This is due to the statistical properties of textures, which render models suitable for processing of cartoon-type natural image to become irrelevant to the processing of fully-textured images. Of special interest are the Natural Stochastic Textures (NST). They depict statistical characteristics that are not encountered in highly structured natural images or in structured textures [5, 6, 7, 8].

NST are considered to be realizations of random processes, since they exhibit certain properties, such as scale-invariance (self-similarity), that are expressed via their statistical properties and not their geometrical properties. In a structured texture, for instance, a line is a self-similar structure in that it retains its shape across scales. In NST, however, a structured area will be similar to its own subset in terms of its variance or functions of higher moments.

The main properties of NST are Gaussianity and statistical self-similarity [6, 7, 9]. Observing that these properties characterise the fractional Brownian motion (fBm), we have previously proposed the fBm as a model suitable for gray-level NST images and color NST images [10]. Some NST, however, contain skeleton-type structures and cannot be fully characterised as realizations of random processes. These structures, while being conceptually similar to edges in a cartoon-type or a general natural image, are not well represented by natural image models. These properties can be validated via the phase information characteristic of an image. Therefore, the Fourier phase should also be incorporated into the NST processing scheme [11].

In this work we investigate the phase structure of textured images. We analyze basic structures

in 1D and 2D signals, identify the magnitude and phase structure of anisotropic textures, and propose a model for the phase of structured textures. We then present an anisotropic texture model and show a non-local algorithm for image deblurring/deconvolution. To incorporate anisotropy into an NST texture model, we propose a combined model for isotropic and anisotropic textures, and show its application in image decomposition to its stochastic and structured ingredients. The combined model can be used for image processing tasks such as deconvolution, super-resolution and others.

2 Background

2.1 Magnitude and phase of NST

It is well-known that in a Fourier representation of images the phase plays an important role, much more so than the magnitude [12]. This is mainly due to the fact that the phase plays a more significant role in the representation of contours, edges and other skeleton-type structures, that are dominant factors in image visual assessment and in recognition. To create an edge in a 1D signal, for instance, all harmonic functions composing the image need to be in the same phase, i.e. “phase-locked”. A small deviation in the phase will affect the edge’s coherence, thus distorting it severely in terms of visual appearance.

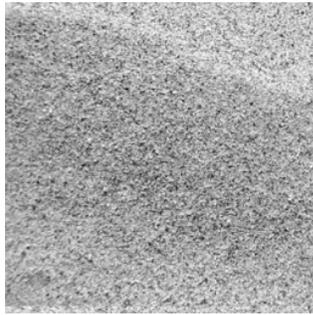
A similar deviation in the magnitude, on the other hand, would alter the edge structure, but since all frequencies will still be in phase, the edge will remain apparent. It is well-known that two images with the same phase will look more similar to one another than two images with the same magnitude [12, 13]. Further, since the characteristic spectrum of a natural image decreases with its frequency, an image with a constant magnitude can be considered to be a high-passed version of the image, enhancing edge-type features, important for shape recognition.

Whereas the phases of natural structured images has been investigated thoroughly, the phase of stochastic textures is usually considered to be less important than its magnitude [14, 15], due to a common assumption that textures can be represented using their autocorrelation, which corresponds to Fourier magnitude [11]. Dividing textures into two classes, structured and stochastic, we indeed observe that in a pure stochastic texture the phase is not important (Fig. 1).

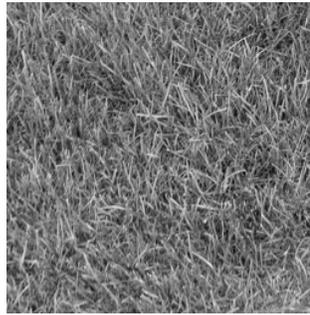
However, textures cannot be completely classified into these two classes; most of the textures contain some stochastic content and some structured content, for which phase is a crucial ingredient. Indeed, some studies exploit the phase in texture reconstruction [11] and synthesis [16].

2.2 Fractional Brownian motion

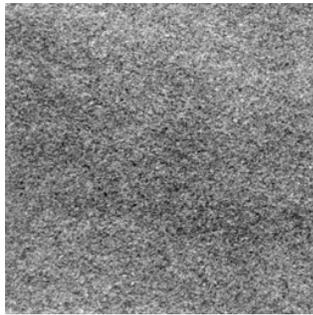
Fractal properties (scale invariance) are found in many natural images and phenomena [17]. The fBm is a well-known fractal process, widely used in the context of analysis of natural images. It is a self-similar Gaussian random process, which was introduced by Mandelbrot and Van-Ness as



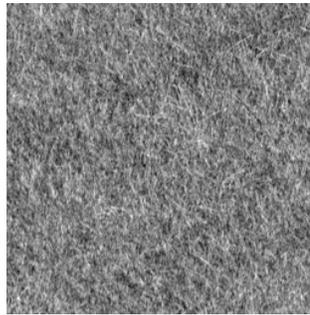
(a) Isotropic texture



(b) Anisotropic texture



(c) Isotropic texture with phase distortion



(d) Anisotropic texture with phase distortion

Figure 1: The importance of phase in textures, expressed via visual assessment of textures with distorted phase: (1a) An isotropic stochastic texture. (1b) an anisotropic texture, containing stochastic as well as structured information. (1c) Result of phase distortion by means of i.i.d Gaussian noise with $\sigma = 1$ (PSNR 14.9dB), applied to (1a). (1d) Result of the same degradation, applied to (1b) (PSNR 18.1dB). We observe that despite the same noise variance applied to both images, the details of the isotropic texture are retained due to its negligible dependency on the Fourier phase. The anisotropic texture, however, contains fine structures and its details are therefore severely distorted with application of the i.i.d Gaussian noise.

a model suitable for natural images [18]. The fBm generalizes the well-known Brownian motion in that the increments are stationary but not independent. It is defined, in one dimension, as a Gaussian process with zero mean and the following autocorrelation function:

$$\mathbb{E}\{B_H(t)B_H(s)\} = \frac{\sigma^2}{2} \left(\|t\|^{2H} + \|s\|^{2H} - 2\|t-s\|^{2H} \right), \quad (1)$$

where $\sigma^2 > 0$ is a known variance, and $H \in (0, 1)$ is the Hurst parameter that characterises the regularity of the process. Lower values of H correspond to rougher signals, and higher values of H correspond to smoother signals. Statistically, the process exhibits negative correlation between samples (anti-persistence) for $H \in (0, 0.5)$, and positive correlation between samples for $H \in (0.5, 1)$. For $H = 0.5$ the fBm degenerates to the standard Brownian motion.

This process exhibits two important properties, known to characterise natural images, in the context of Mandelbrot's work on fractals [18]. The first one is long-range dependencies (LRD) between samples, where for $H \in (0.5, 1)$ the sum of the correlations of the increments diverges [19]. The second property of the fBm is its fractal (self-similarity) property:

$$B_H(at) \stackrel{d}{=} \|a\|^H B_H(t),$$

for $a > 0$, where the equality is in distribution. This equality indicates that the sample distribution across different scales is varied only by a constant depending on the scale, a , and on H . These two properties highlight the relevance of this process to natural textures, as the latter often exhibit similarities between adjacent as well as distant pixels.

The fBm does not have a unique 2D extension [20]. In this study we choose the straightforward manner in which the process indices in the definition of the covariance (1) are extended to 2D:

$$t \rightarrow (t_1, t_2), \quad s \rightarrow (s_1, s_2).$$

The $\|\cdot\|$ operator is then the L_2 norm of a point in 2D space. The 2D fBm is governed by a single Hurst parameter and is, therefore, isotropic. We note that there are various extensions of the fBm to multi-fractal processes, in which the fractal dimension is varied in space or in frequency [21, 22, 23].

A useful process, derived from the fBm, is the fractional Gaussian noise (fGn), defined as the first order difference of the fBm:

$$W_H(t) = B_H(t) - B_H(t-1).$$

The fGn is a stationary process exhibiting asymptotical self-similarity, and it has the same memory properties as the fBm w.r.t H : short-range dependence is exhibited for $H \in (0, 0.5)$ and LRD is exhibited for $H \in (0.5, 1)$. In the case of $H = 0.5$, the fGn degenerates to white Gaussian noise (as the increment of the Brownian motion).

The two aforementioned processes, fBm and fGn, are useful in texture processing and analysis, due to the fact that many processes exhibit Gaussianity and self-similarity [9]. It can be shown that the fBm is the only process (in 1D) to exhibit these two properties, considering only processes with stationary increments. In this study we address fBm/fGn patches in a discrete framework, in which the covariance is assumed to be the discrete covariance. We note that there are extended versions of the discrete fBm [24]. The limited patch size raises the question of whether or not fractal behaviour could be assessed using small patches; this is addressed in Sec. 2.3.

2.2.1 Karhunen-Loève (KL) decomposition of fBm

An immediate representation of a discrete fBm/fGn signal, \underline{f} , shall be $\underline{f} = W\underline{x}$, where \underline{x} is an i.i.d Gaussian noise and WW^T is the fBm/fGn covariance matrix. However, this method requires the calculation of the square root of the covariance matrix and does not provide any insight as to the stability of its inverse (as is required in likelihood functions used in this work). Therefore, we present basic KL results that will be used in the following sections.

Let $f(x, y)$ be an fBm/fGn image. Using the Karhunen-Loève transform [25], $f(x, y)$ can be represented as the infinite sum $f(x, y) \stackrel{d}{=} \sum_{i=0}^{\infty} \xi_i \psi_i(x, y)$, where $\{\xi_i\}$ are uncorrelated standard Gaussians, and $\psi_i(x, y)$ are the eigenfunctions of the covariance of $f(x, y)$. Since in all practical cases we address discrete signals, we use the discrete formulation, in which \underline{f} is a vectorized image matrix. The eigenfunctions are then obtained by the solution of the following equation:

$$R_f \psi'_i = \lambda_i \psi'_i, \quad (2)$$

where R_f is the covariance matrix of \underline{f} , the vectorized version of $f(x, y)$, and ψ'_i is a vectorized function. The solution to the complete set of ψ'_i is then obtained from (2) as follows:

$$R_f \Psi = \Psi \Lambda,$$

where Λ is a diagonal matrix with $\{\lambda_i\}$ on its diagonal, and Ψ is a matrix whose columns are the corresponding functions, ψ'_i . Ψ and Λ can, therefore, be obtained by eigendecomposition of the covariance matrix, R_f . The resulting columns of Ψ represent the spanning functions in decreasing order of signal energy content, and \underline{f} is obtained by the following sum:

$$\underline{f} \stackrel{d}{=} \sum_{i=1}^K \xi_i \sqrt{\lambda_i} \psi'_i, \quad (3)$$

where $\{\xi_i\}$ are uncorrelated standard normal variables. The summation (3) can be represented in matrix form as follows:

$$\underline{f} \stackrel{d}{=} \Psi \Lambda^{0.5} \xi \triangleq \Phi \xi, \quad (4)$$

where $\underline{\xi} = (\underline{\xi}_1, \dots, \underline{\xi}_K)^T$. We note that while $\Phi\Phi^T$ is the fBm covariance, this case is different than obtaining any square root of a matrix, since the vectors in Φ are organized by descending order of energy contribution of the signal. Further, the basis functions, Ψ , have known analytic expressions only in the case of $H = 0.5$, in which the fBm coincides with the standard Brownian motion [26].

An implication of this decomposition, used in this study, is that if R_f is non-invertible, an inverse matrix can, nevertheless, be obtained using almost all eigenvectors with minimal loss in fidelity to the true covariance matrix.

2.3 Fractal behaviour on discrete and limited supported patches

In many image processing applications a patch size of 3×3 or 5×5 is used for modeling purposes. Considering the classical image model of smooth surfaces and edges, it is clear that one can detect an edge even in these small patch sizes, as well as detect a smooth area. This is due to the very high dimension endowed by 9 or 25 pixels with gray-level values in the range of $[0, 255]$. However, in order to model fractal behaviour, the patch size should be larger. The fractal property of scale invariance requires sufficient patch size in order to assess the variance in different scales; for this task, the sample size should be larger, as we show hereinafter. Nevertheless, we show that fractal behaviour can indeed be estimated in moderately sized patches as used in this work and others.

Let $\{X(n), 0 \leq n < N\}$ be a 1D-fBm defined on a discrete grid. $X(n)$ is a discrete Gaussian process and $X(cn) \stackrel{d}{=} c^H X(n)$ for $c \in \mathbb{N}^+$. The process covariance is $R_X(n, m) = \mathbb{E}\{X_n X_m\} = \sigma^2 (n^{2H} + m^{2H} - |n - m|^{2H})$.

Let $W_\tau(n)$ denote the τ th order increment of fBm:

$$W_\tau(n) \triangleq X(n + \tau) - X(n).$$

$W_\tau(n)$ is stationary (due to the stationary increment property of fBm) and has the following variance and covariance:

$$R_{W_\tau}(n, n) = \mathbb{E}\{(X_{n+\tau} - X_n)^2\} = \mathbb{E}\{(X_\tau - X_0)^2\} \quad (5)$$

$$= \mathbb{E}\{X_\tau^2\} = R_X(\tau, \tau) = \sigma^2 (\tau^{2H} + \tau^{2H}) = 2\sigma^2 \tau^{2H} \quad (6)$$

$$R_{W_\tau}(n, n + k) = R_{W_\tau}(0, k) = \mathbb{E}\{(X_\tau - X_0)(X_{\tau+k} - X_k)\} = \mathbb{E}\{X_\tau(X_{\tau+k} - X_k)\} \quad (7)$$

$$= R_X(\tau, \tau + k) - R_X(\tau, k) \quad (8)$$

$$= \sigma^2 (\tau^{2H} + |\tau + k|^{2H} - \tau^{2H} - k^{2H} - |k|^{2H} + |\tau - k|^{2H}) \quad (9)$$

$$= \sigma^2 (|\tau + k|^{2H} + |\tau - k|^{2H} - 2|k|^{2H}). \quad (10)$$

Therefore, $V(\tau)$, the log-variance/scale plot as a function of τ is:

$$V(\tau) \triangleq \log(R_{W_\tau}(n, n)) = \log(2\sigma^2) + 2H \log \tau, \quad (11)$$

and can be used to estimate H . This stems from successful estimation of the increment variances, $R_{W_\tau}(n, n)$, which converges to the true variance for high sample sizes. We would like to estimate a minimal sample size, N , from which fractal behaviour can be detected. Considering an unknown patch of size $N = n \times n$, we would like to decide whether was generated by an i.i.d noise source (WGN) or by fractal behaviour. We use i.i.d noise as a prototype to the simplest random texture that might be confused with fBm in case of small sample size.

Performing an hypothesis testing scheme to obtain a threshold for rejection of the white noise hypothesis requires knowledge of the probability of a suitable test statistic. In our case, a suitable test statistic will be the variance of the increments; while the variance of the increments of fGn of order τ is given by (6), the variance of the increments of i.i.d Gaussian noise, denoted by W_τ^g , is given by:

$$R_{W_\tau^g}(n, n+k) = \mathbb{E}\{(X_{\tau+n} - X_n)(X_{\tau+n+k} - X_{n+k})\} \quad (12)$$

$$= \mathbb{E}\{(X_\tau - X_0)(X_{\tau+k} - X_k)\} \quad (13)$$

$$= R_X(\tau, \tau+k) - R_X(0, \tau+k) - R_X(\tau, k) + R_X(0, k) \quad (14)$$

$$= R_X(0, k) - R_X(0, \tau+k) - R_X(0, k-\tau) + R_X(0, k) \quad (15)$$

$$= \sigma^2(2\delta(k) - \delta(\tau+k) - \delta(\tau-k)), \quad (16)$$

and

$$R_{W_\tau^g}(n, n) = 2\sigma^2.$$

Therefore, the variance of the increments can serve as a suitable statistic for detecting fBm versus white noise. Since both processes are Gaussian, this quantity is the simplest statistic to use, as the fractal property cannot be assessed via first-order statistics.

A complete hypothesis testing scheme requires knowing the probability law of the test statistic. In many cases, one uses approximations for the probability law for large number of samples. The F-test, for instance, is a commonly used test in the case of regression coefficient fitting [27]. In our case, however, we would like to compare the two hypotheses in for a small number of samples, which renders asymptotical approximations to be impractical. Further, we extract regression coefficients using a series of increment variances, that are inherently dependent even in the case of i.i.d noise (16), and therefore the independence assumption is not valid here.

The sample size, N , affects the quality of estimating $R_{W_\tau}(n, n)$. We would, therefore, like to use the sample variance $\hat{R}_{W_\tau}(n, n)$ as a test statistic, which is dependent on N . As the sample variance is of a correlated Gaussian sequence (16), it does not have a known form [27]. Instead, we

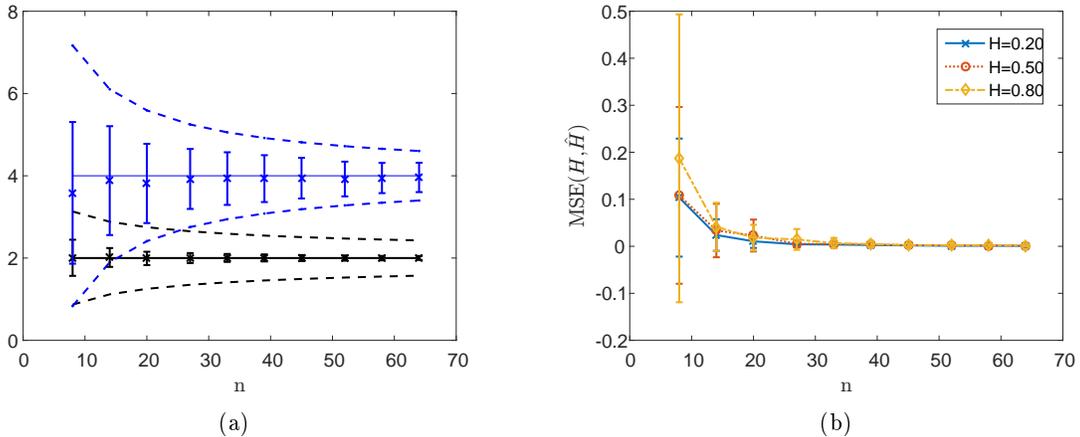


Figure 2: Assessing minimal patch size for fractal behaviour. (2a) The variance of the sample variance: dashed lines indicate analytical results, markers and solid lines depict means and standard deviations of experiments of various sample sizes. The variances of the sample variance of fBm increments with $H = 0.5$ (depicted in thin blue lines) are compared with the same statistical function of i.i.d noise sequences (depicted in thick black lines). We observe that a patches of size smaller than 10×10 are not sufficient for distinction between white noise and fractal images. (2b) An Hurst parameter estimation experiment, further indicating that a sample size of less than 10×10 is insufficient for satisfactory estimation.

evaluate only the variance of the sample variance for a general correlated sequence (Appendix A).

Assuming σ^2 is known, $\hat{R}_{W_T^g}(n, n)$, an estimator for $R_{W_T^g}(n, n)$ has a mean $2\sigma^2$ and standard deviation as given by (53), a function of the sample size, N . Comparing the variance for the sample variance in the two cases of white noise versus fBm (Fig. (2a); solid lines depict the estimator means and dashed lines indicate standard deviations), we observe that patch sizes commonly used for natural images with lower dimensions than 8×8 are not enough to detect fractal behaviour, whereas patch sizes of 16×16 and 32×32 are more probable to be differentiated from i.i.d noise (Fig. 3). We note, however, that this analysis refers to a local processing of patches, done independently on other patches. To complement this experiment, H -estimation was performed for different patch sizes (Fig. 2b), indicating that a patch size of 16×16 is sufficient for an MSE of less than 0.1.

3 Phase analysis for 1D and 2D signals

3.1 The phase of one-dimensional signals

In 1D signals, the geometric shapes considered are edges, rectangular windows, signals with linear slopes (ramps), and noise. The former three types correspond to some coherence in terms of the phase, whereas the latter corresponds to random phase. While the goal of phase analysis is 2D images, in which there is true textural anisotropy, we consider this case in order to show some

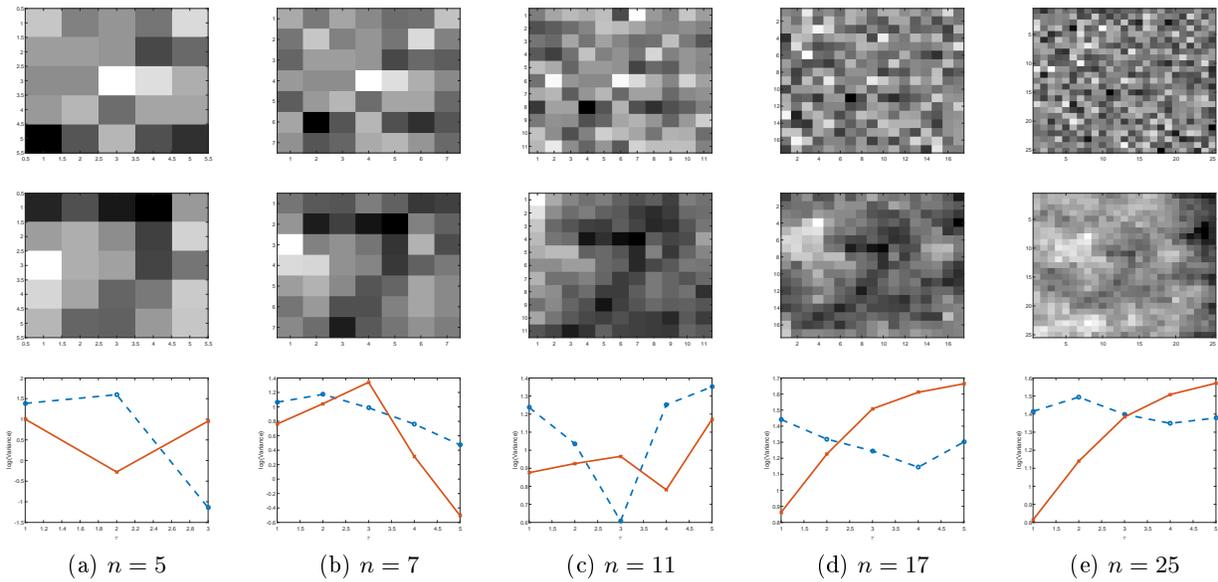


Figure 3: Example for various patch sizes and their fractal behaviour. Each column shows a realization of white noise (first row), fBm with $H = 0.5$ (second row) and their increment log-variance of order τ (third row), shown in dashed-blue and solid-red for the white noise and fBm, respectively. The R^2 values for the white noise were (in increasing patch size excluding the smallest patch size) 0.68, 0.01, 0.56 and 0.34 and the R^2 values for the fBm were 0.36, 0.18, 0.97 and 0.98. High values of R^2 are obtained for large patch sizes, whereas for small patch sizes the fractal surface is visually and statistically indistinguishable from white noise.

principal phase-based properties.

Let $x_e(t)$ be an edge defined on a limited support, $-T < t < T$, as follows:

$$x_e(t) = \begin{cases} 1, & 0 < t < T \\ -1, & -T < t \leq 0 \\ 0, & \text{otherwise,} \end{cases}$$

and let us calculate its phase:

$$\begin{aligned} \hat{x}_e(\omega) &= \int_{-T}^T x_e(t) e^{-j\omega t} dt \\ &= \int_{-T}^T x_e(t) \cos(\omega t) dt - j \int_{-T}^T x_e(t) \sin(\omega t) dt \\ &= -2j \int_0^T 1 \cdot \sin(\omega t) dt = 2j \left[\frac{1}{\omega} \cos(\omega t) \right]_0^T \\ &= 2j \frac{1}{\omega} (\cos(\omega T) - 1) \end{aligned}$$

we obtain a purely imaginary Fourier transform, and therefore a constant phase, indicating that all the frequency component are required to be in the same phase to create the edge. Let $x_r(t)$ denote a rectangular centered window:

$$x_r(t) = \begin{cases} 1, & -c < t < c \\ 0, & \text{otherwise,} \end{cases}$$

where $c < T$. A similar analysis yields:

$$\begin{aligned} \hat{x}_r(\omega) &= \int_{-T}^T x_r(t) e^{-j\omega t} dt \\ &= \int_{-T}^T x_r(t) \cos(\omega t) dt - j \int_{-T}^T x_r(t) \sin(\omega t) dt \\ &= \int_{-c}^c 1 \cdot \cos(\omega t) dt = 2 \int_0^c \cos(\omega t) dt \\ &= 2 \frac{1}{\omega} (\sin(\omega c) - 1). \end{aligned}$$

The Fourier transform is purely real, indicating a constant phase in this case as well.

Let $x_s(t)$ denote a constant slope signal (a ramp):

$$x_s(t) = \begin{cases} t, & -T < t < T \\ 0, & \text{otherwise,} \end{cases}$$

with the following Fourier transform:

$$\begin{aligned}\hat{x}_s(\omega) &= \int_{-T}^T t e^{-j\omega t} dt \\ &= -j \int_{-T}^T t \sin(\omega t) dt.\end{aligned}$$

While the calculation requires extra steps, we observe the same phenomenon. We can conclude by showing that symmetry (including anti-symmetry) is a sufficient conditions for a constant phase. The versions of edges or rectangular windows we discussed are all special cases of this statement: let $x_q(t)$ be a symmetric (or anti-symmetric) signal with symmetric support, defined with the parameter $q = \pm 1$ as follows:

$$x_q(t) = \begin{cases} q \cdot x_q(-t), & -T < t < T \\ 0, & \text{otherwise,} \end{cases}$$

$\hat{x}_q(\omega)$, the Fourier transform of $x_q(t)$, is given by:

$$\begin{aligned}\hat{x}_q(\omega) &= \int_{-T}^T x_q(t) e^{-j\omega t} dt \\ &= \int_{-T}^0 x_q(t) e^{-j\omega t} dt + \int_0^T x_q(t) e^{-j\omega t} dt \\ &= \int_0^T x_q(-t) e^{j\omega t} dt + \int_0^T x_q(t) e^{-j\omega t} dt \\ &= \int_0^T x_q(t) [q e^{j\omega t} + e^{-j\omega t}] dt \\ &= \int_0^T x_q(t) [q \cos(\omega t) + qj \sin(\omega t) + \cos(\omega t) - j \sin(\omega t)] dt \\ &= \int_0^T x_q(t) [\cos(\omega t) (q + 1) + j \sin(\omega t) (q - 1)] dt.\end{aligned}$$

Since $\cos(\omega t) (q + 1) + j \sin(\omega t) (q - 1)$ is either real or imaginary, we conclude that $x_q(t)$ has a constant phase, either 0 or π .

Further, let $y(t) = x(t + t_0)$ be a shifted version of any symmetric (or anti-symmetric) signal, $x(t)$. Without loss of generality, assume $x(t)$ is symmetric and has, therefore, zero phase. Using known Fourier transform properties, we have:

$$\begin{aligned}\hat{y}(\omega) &= e^{j\omega t_0} \hat{x}(\omega) \\ \angle \hat{y}(\omega) &= \angle e^{j\omega t_0} + \angle \hat{x}(\omega) = \omega t_0.\end{aligned}$$

We observe that a shift of a symmetric signal causes a significant change in the phase of the signal. A shift is, in general, only a minor modification that has little visual effect; consider, for instance, a slightly shifted patch of an edge segment in an image. We see that the zero-phase property of edges is not shift invariant. Further, the amount of shift, t_0 , also affects the phase structure.

As a conclusion from this analysis, let us consider the histograms of two signals, $y(t)$ and $z(t)$. $y(t)$ is a shifted symmetric signal, as defined earlier, and $z(t)$ is a zero-mean, unit variance, white Gaussian noise (WGN). The Fourier transform of $z(t)$ is given by:

$$\begin{aligned}\hat{z}(\omega) &= \int_{-T}^T z(t) e^{-j\omega t} dt \\ &= \int_{-\infty}^{\infty} z(t) h(t, \omega) dt,\end{aligned}$$

where

$$h(t, \omega) = \mathbf{1}_{t \in [-T, T]} e^{-j\omega t}.$$

Since $z(t)$ is WGN, passing through the linear system $h(t, \omega)$ yields a complex Gaussian signal. Denote

$$\begin{aligned}\hat{z}_r(\omega) &= \Re \hat{z}(\omega) = \int_{-T}^T z(t) \cos(\omega t) dt \\ \hat{z}_i(\omega) &= \Im \hat{z}(\omega) = \int_{-T}^T z(t) \sin(\omega t) dt\end{aligned}$$

we observe that

$$\begin{aligned}\text{Cov}\{\hat{z}_r(\omega), \hat{z}_i(\omega)\} &= \mathbb{E}\{\hat{z}_r(\omega) \hat{z}_i(\omega)\} \\ &= \mathbb{E}\left\{\int_{-T}^T z(t) \cos(\omega t) dt \int_{-T}^T z(t) \sin(\omega t) dt\right\} \\ &= \mathbb{E}\left\{\int_{-T}^T \int_{-T}^T z(s) z(t) \sin(\omega s) \cos(\omega t) ds dt\right\} \\ &= \int_{-T}^T \int_{-T}^T \delta(s-t) \sin(\omega s) \cos(\omega t) ds dt \\ &= \int_{-T}^T \sin(\omega t) \cos(\omega t) dt = 0,\end{aligned}$$

and the two counterparts are, therefore, uncorrelated for each ω (since this is a Gaussian process, they are also independent), and the marginal distributions have zero mean. We are interested in the phase histogram. Let A_z be a random variable defined as follows:

$$A_z(\omega) = \arctan\left(\frac{\hat{z}_i(\omega)}{\hat{z}_r(\omega)}\right).$$

The ratio $\frac{\hat{z}_i(\omega)}{\hat{z}_r(\omega)}$ is a normal ratio. The variances of the two variables are different, but there exist some constant $\sigma > 0$ so that $R = \frac{\hat{z}_i(\omega)}{\hat{z}_r(\omega)} \stackrel{d}{=} \frac{\sigma_i X}{\sigma_r Y} \triangleq \sigma Z$ where $X, Y \sim \mathcal{IN}(0, 1)$. Therefore, $Z \sim \text{Cauchy}(0, 1)$, and $R \sim \text{Cauchy}\left(0, \frac{\sigma_i}{\sigma_r}\right)$. Explicitly,

$$\begin{aligned}
\sigma_i^2(\omega) &= E \left[\int_{-T}^T z(t) \sin(\omega t) dt \int_{-T}^T z(s) \sin(\omega s) ds \right] - E[\hat{z}_i(\omega)]^2 \\
&= E \left[\int_{-T}^T \int_{-T}^T z(t) z(s) \sin(\omega t) \sin(\omega s) dt ds \right] \\
&= \int_{-T}^T \sin(\omega t)^2 dt = \frac{1}{2} \int_{-T}^T 1 - \cos(2\omega t) dt \\
&= T - \int_0^T \cos(2\omega t) dt = T - \frac{1}{2\omega} \sin(2\omega T) \\
\sigma_r^2(\omega) &= \frac{1}{2} \int_{-T}^T 1 + \cos(2\omega t) dt = T + \frac{1}{2\omega} \sin(2\omega T)
\end{aligned}$$

Therefore,

$$\sigma^2(\omega) = \frac{T - \frac{1}{2\omega} \sin(2\omega T)}{T + \frac{1}{2\omega} \sin(2\omega T)}.$$

The cdf for R is:

$$F_R(x) = \frac{1}{\pi} \arctan\left(\frac{x}{\sigma(\omega)}\right) + \frac{1}{2}.$$

A histogram bin B_j is defined as:

$$B_j = \sum_i \mathbf{1}_{A_z(\omega_i) \in [b_{j-1}, b_j]},$$

where $\{b_1, \dots, b_m\}$ denote the bin limits of a histogram with $m - 1$ bins. The expected value of B_j is given by:

$$\mathbb{E}\{B_j\} = \sum_i \mathbb{E}\left\{\mathbf{1}_{A_z(\omega_i) \in [b_{j-1}, b_j]}\right\} \quad (17)$$

$$= \sum_i \mathbb{P}\{A_z(\omega_i) \in [b_{j-1}, b_j]\} \quad (18)$$

$$= \sum_i \mathbb{P}\{R(\omega_i) \in [\tan b_{j-1}, \tan b_j]\} \quad (19)$$

$$= \frac{1}{\pi} \sum_i \left[\arctan\left(\frac{\tan b_j}{\sigma(\omega_i)}\right) - \arctan\left(\frac{\tan b_{j-1}}{\sigma(\omega_i)}\right) \right]. \quad (20)$$

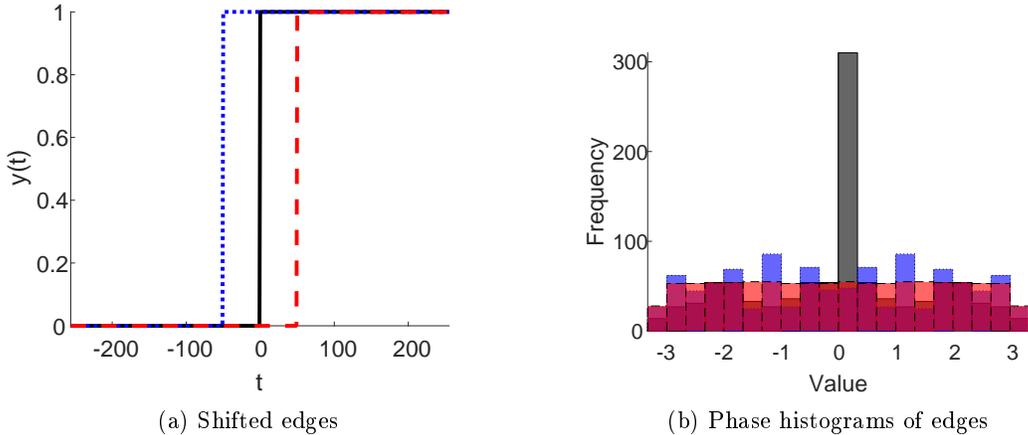


Figure 4: Effects of shifts on the phase histogram of edge signals. (4a) A symmetric edge (solid black) and two shifted edges (dotted blue and dashed red), with signal length of 512 samples. (4b) 20-bin histograms of the phase of the depicted edges. Each edge signal’s histogram is shown in the same color as the edge signal. While a symmetric edge has an approximately zero phase response, indicated by a peak in the zero bin, the shifted versions have significantly varied histograms, more distinctive of white noise.

To calculate the variance, first consider that for $i \neq k$, $A_z(\omega_i)$ and $A_z(\omega_j)$ are independent, as functions of the independent variables $\hat{z}(\omega_i)$ and $\hat{z}(\omega_j)$, respectively, as follows from the properties of the Fourier transform and Gaussianity. Therefore, the variance of B_j is the sum of variances of each indicator can be represented using (20).

It can be confirmed by a deterministic simulation that both the mean and variance of B_j are approximately independent of j , the bin location. As a practical example for a 20-bin histogram, consider a noise sequence, $z(t)$, with $T = 100$, with frequency discretization to $w = 1024$ frequency components. Applying the formulas above, we obtain $\sigma^2(\omega) \approx 1$ with sample standard deviation (w.r.t ω) 32.9×10^{-5} , $\mathbb{E}\{B_j\} \approx 51.2$ with sample standard deviation 18.7×10^{-5} and $\text{Var}(B_j) \approx 48.64$ with sample standard deviation 16.9×10^{-5} . The sample standard deviations decreases with the number of histogram bins, but we observe that a practical number of bins is enough for approximate independence of the bin statistics to its location, j .

The conclusion from this derivation is the somewhat intuitive result that the phase histogram of the noise signal, $z(t)$, is approximately constant, indicative of a uniform distribution. On the other hand, the phase histogram of the shifted symmetric signal, $y(t)$, will also be approximately constant, inasmuch as the phase is linear (modulo 2π). Therefore, the phase histogram cannot distinguish even between these two simple cases (Fig. 4).

Another observation for the description of edges in terms of the statistical structure of their phase is as follows. While the histogram of the phase itself is not enough to distinguish between $z(t)$ and $y(t)$, the phase’s spatial derivative will be provide important information; the derivative of the

phase of $z(t)$ will still be random with a uniform histogram, whereas the corresponding derivative of $y(t)$ will be constant in all places other than several discontinuous points caused by the modulo operation. The spatial phase derivatives, may, therefore, serve as a tool for edge detection from phase.

3.1.1 Magnitude and phase relationships

Let $x(t)$ be a one-dimensional signal with its Fourier transform, $\hat{x}(\omega)$. In the case where $x(t)$ is an edge, we have an analytic expression for $\hat{x}(\omega)$, and the exact reconstruction of the edge requires all Fourier coefficients. However, in many practical cases, not all coefficients are required for satisfactory reconstruction of a signal. That is, the signal can be reconstructed via only a subset of its coefficients with negligible visual difference.

Assume the frequency indices $\{\omega_i\}$ can be divided to two distinct subsets, $\{v_i\}$ and $\{u_i\}$, indicating “strong” and “weak” frequency coefficients, respectively, so that $\{v_i\} \cup \{u_i\} = \{\omega_i\}$ and $\{v_i\} \cap \{u_i\} = \emptyset$. A method for partitioning $\{\omega_i\}$ into these two subsets is based on a magnitude threshold, t , so that $\omega_i \in \{v_i\} \iff |\hat{x}(\omega_i)| > t$, where t is chosen so that most of the signal’s energy is retained considering only the frequencies in $\{v_i\}$.

Empirically, we observe that the phases of coefficients from frequencies in $\{u_i\}$ tend to be independent and uniform even if the signal itself has some structural information. Weak frequencies in magnitude tend to have low energy, a property that can be evaluated easily. In the case of phases, however, a weak coefficient will tend to exhibit no dependence with other coefficients and have uniform marginal distribution. A possible explanation for this phenomenon is as follows: if $|\hat{x}(\omega_k)| = \sqrt{\hat{x}_r^2(\omega_k) + \hat{x}_i^2(\omega_k)}$ is a small enough magnitude (where subscripts r and i indicate real and imaginary parts, respectively), we have that

$$\angle \hat{x}(\omega_i) = \arctan\left(\frac{\hat{x}_i(\omega_k)}{\hat{x}_r(\omega_k)}\right).$$

Since the argument of the arc-tangent is a quotient of two small coefficients, there is very little dependence in the true values and we obtain unstable, or unexpected, behaviour, which is observed empirically as uniform distribution over all possible phase values. These properties are hard to detect, since they require some joint filtering of a large group of coefficients. The weak coefficient’s phases act as noise when performing phase-based analysis of all coefficients, and should, therefore, be ignored.

3.2 The phase of two-dimensional signals

An anisotropic signal exhibits, by definition, different behaviour in different orientations. This behaviour cannot be captured by a one-dimensional signal. Therefore, while 1D analysis is simpler, we must extend it to two dimensions. We will focus in the analysis of the local phase.

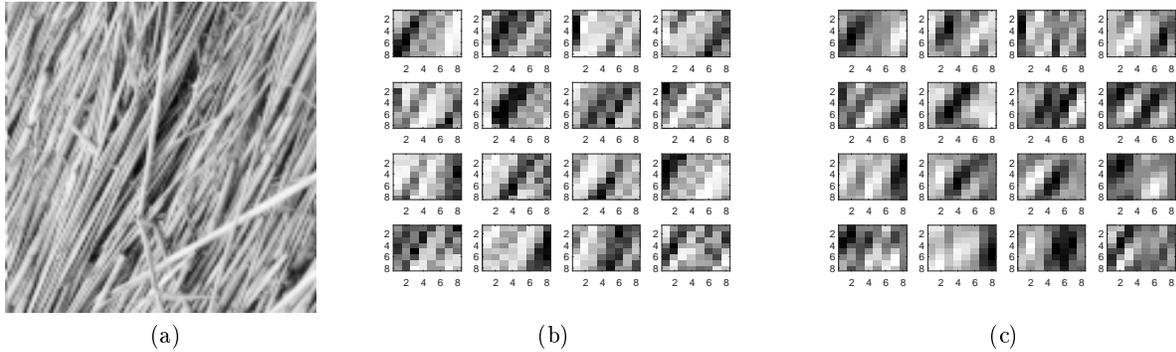


Figure 5: Discarding nearly all frequency components of patches (5b) from an anisotropic image (5a) yields patches with similar visual appearance (5c).

As a simple example, consider the following signal:

$$f(x, y) = 2 [\cos(ax + by) + \cos(cx + dy)]$$

with its Fourier transform,

$$F(u, v) = \delta(u - a, v - b) + \delta(u + a, v + b) + \delta(u - c, v - d) + \delta(u + c, v + d).$$

$f(x, y)$ is anisotropic, and $F(u, v)$ is real, indicating zero phase. Further, any generalization of $f(x, y)$, in the form of

$$g(x, y) = \sum_{i=1}^N \alpha_i \cos(a_i x + b_i y),$$

where $\{\alpha_i\}$, $\{(a, b)_i\}$ and N are known, will have the same properties. Therefore, non-zero phase is not a necessary condition for anisotropy: adding a random phase to each of the components of $g(x, y)$ will change their relative phase, but the anisotropy will be retained, as the same frequency components (the location of the delta functions in frequency) are retained.

As an example, consider an anisotropic texture image (Fig. 5a). While this image does not contain edges in the usual sense - as boundaries between objects, it does exhibit high frequencies. Decomposing into 8×8 patches (Fig. 5b depicts a random selection), we observe that by reconstructing the patches using the 4 frequency components with the highest magnitude, the anisotropy of the patch is retained.

While a model for the local phase might be obtained, it would need to model a small number of coefficients for each patch, corresponding to the highest magnitudes. In this report, however, we focus on a different aspect of the phase structure. We observe that a patch may be anisotropic as well

as contain strong local frequencies (with high magnitudes), sufficient for description of anisotropic textures, even when the phase is randomized (and uniformly distributed).

Let $g_{\theta_i}(x, y)$ be the following patch-based model:

$$g_{\theta}(x, y) = \sum_{i=1}^N \alpha_i \cos(a_i x + b_i y + \phi_i), \quad (21)$$

for a known parameter vector $\theta = (N, \{\alpha_i, a_i, b_i, \phi_i\}_i)$. Consider two patches, P_1 and P_2 , with their corresponding parameter vectors, θ^1 and θ^2 . If the texture is visually homogeneous, it is reasonable to assume that their strong frequency magnitudes, $\{a_i, b_i\}$ will be similar. However, their phase components, $\{\phi_i\}$, will not. Consider the 1D case with a single component:

$$f(t) = \sin(at), \quad 0 < t < T,$$

where $T > \frac{3}{2}$, and consider two patches of length 1, $P_1(t)$ and $P_2(t)$, defined as follows:

$$\begin{aligned} P_1(t) &= f(t), \quad 0 < t < 1 \\ P_2(t) &= f(t), \quad \frac{1}{2} < t < \frac{3}{2}. \end{aligned}$$

While both patches will contain the same frequency component, the different support of each patch dictates its phase; while $P_1(t) = \sin(at)$,

$$\begin{aligned} P_2(t) &= \sin(at), \quad \frac{1}{2} < t < \frac{3}{2} \\ &= \sin\left(a\left(t + \frac{1}{2}\right)\right), \quad 0 < t < 1 \\ &= \sin\left(at + \frac{1}{2}a\right), \quad 0 < t < 1. \end{aligned}$$

We observe that while the frequency magnitude of $P_1(t)$ and $P_2(t)$ are equal, the difference in the spatial domain is due to their difference in phase. A typical image patch will naturally contain more frequency components, but this phenomenon will be approximately retained. It is, therefore, useful to consider the magnitude and phase components separately when analyzing and processing.

Consider $P_0(x, y) = g_{\theta}(x, y)$ to be a prototype patch with several frequency components (21), whose origin coincides with the image origin, $t = 0$. The phase shift for a general patch,

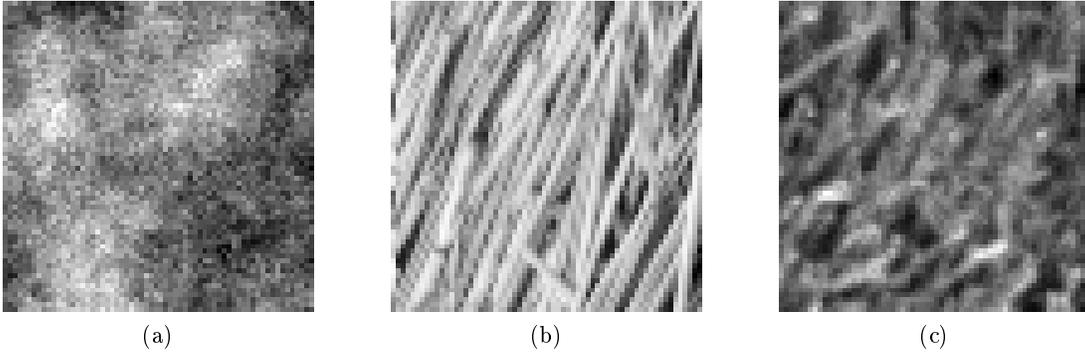


Figure 6: Typical images: (6a) Isotropic texture, (6b) anisotropic texture and (6c) asotropic texture.

$P_{x_0, y_0}(x, y) = g_\theta(x + x_0, y + y_0)$, shall be:

$$\begin{aligned} g_\theta(x + x_0, y + y_0) &= \sum_{i=1}^N \alpha_i \cos(a_i x + a_i x_0 + b_i y + b_i y_0 + \phi_i) \\ &= \sum_{i=1}^N \alpha_i \cos(a_i x + b_i y + \phi_i + h_i(x_0, y_0)), \end{aligned}$$

where

$$h_i(x_0, y_0) = a_i x_0 + b_i y_0 \tag{22}$$

is the local phase shift. Therefore, for each frequency index, i , considered in the patch representation, the phase shift is a linear function (that can be modulo 2π as well) with respect to the patch coordinates, (x_0, y_0) .

The phase function (22) only applies to frequencies propagating throughout the texture. Frequencies with low magnitudes are expected to have less structured non-local phase. It is, therefore, required to decide which frequencies to consider as globally strong frequencies that will satisfy the requirement presented by $h_i(x_0, y_0)$ for a given patch, $P_{x_0, y_0}(x, y)$. In this work we use a threshold for the frequency coefficients' magnitude.

We demonstrate this phenomenon for three types of typical images (Fig. 6): an isotropic texture, an anisotropic texture and an asotropic texture (i.e. a texture not exhibiting isotropy or anisotropy). We analyze the phase by dividing the image into overlapping patches of size 8×8 and calculating the phase of the largest frequencies, for adjacent patch locations (Fig. 7). We observe that for the structured images – anisotropic and asotropic – the frequencies approximately correspond to the phase model (22), whereas the isotropic texture exhibits random phase. The same behaviour can be observed in 2D as well (Fig. 8). The anisotropic and asotropic texture phases depict a surface similar to the surface implied by the phase model (22), up to modulo 2π .

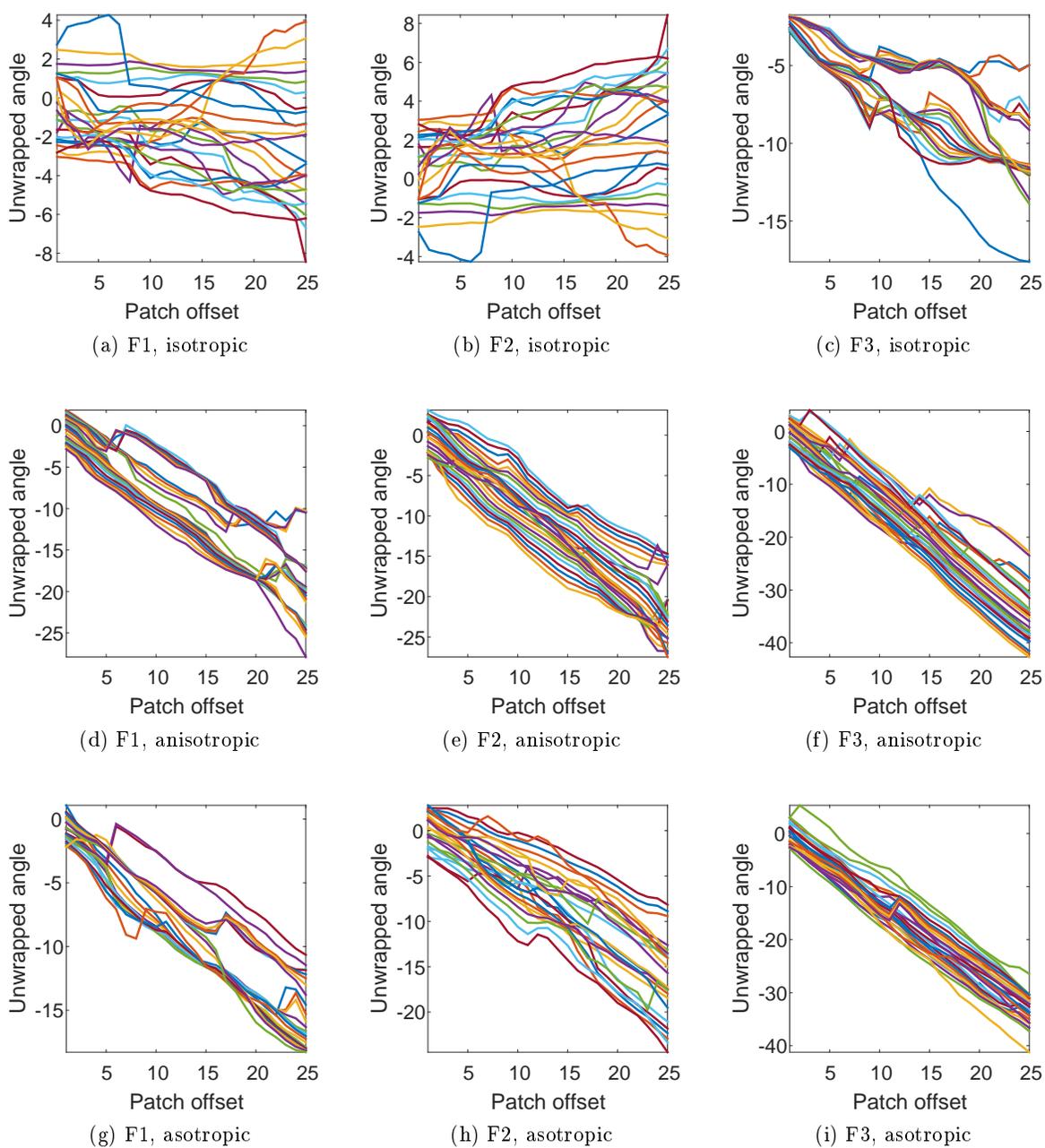


Figure 7: The phase progression of strongest frequencies throughout patch location. The first, second and third rows correspond to the isotropic, anisotropic and asotropic images (Figs. 6a, 6b and 6c), respectively. The first, second and third columns (F1, F2 and F3) correspond to the frequencies with the largest magnitudes, in ascending order. Each figure shows the progression of the unwrapped phase for a patch with a certain offset. Each line depicts the progression of a patch moving from top to bottom. The different lines correspond to different horizontal locations in the image. Anisotropic and asotropic textures exhibit a phase much closer to linear, due to the similar structure in adjacent locations. Isotropic texture exhibits close to random phase.

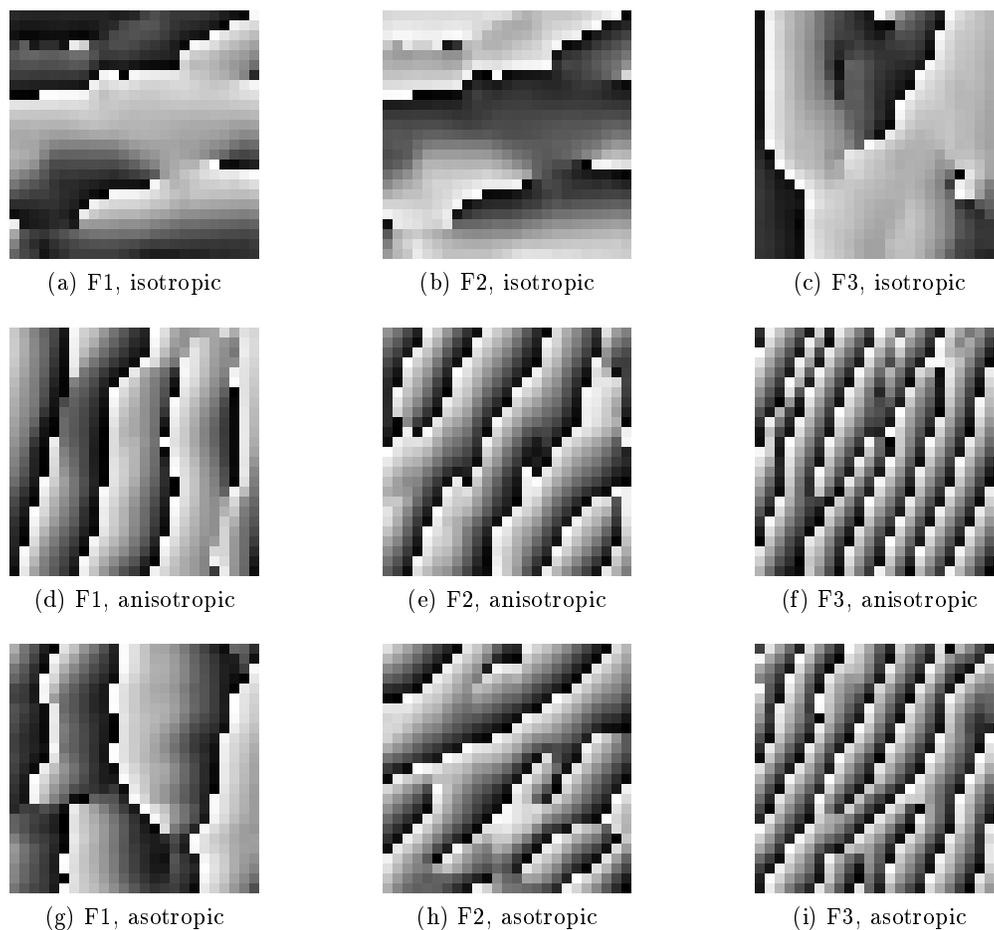


Figure 8: The phases of strongest frequencies throughout 2D patch location. The rows are defined similarly to Fig. 7. Each point in each figure shows the phase of a certain frequency in a patch whose center is located at that point. Similarly to Fig. 7, the anisotropic and asotropic textures exhibit a phase much closer to linear, especially in the second and third strongest frequencies, due to the similar structure in adjacent locations.

To demonstrate the applicability of the Fourier-based model we perform reconstruction of the three images (Fig. 6) using a different number of globally strongest frequencies for reconstruction. Each image is reconstructed via 3, 5 or 10 frequency components per 8×8 patch (Figs. 9 and 10). The anisotropic and isotropic textures are successfully reconstructed even with a low number of frequency components. The isotropic texture is not reconstructed successfully even using a relatively high number of frequency components.

We note that in anisotropic textures, the magnitude plays a smaller role than the phase. Evidently, reconstructing the anisotropic texture (Fig. 6b) by the aforementioned model can be performed using the global frequencies without any magnitude information (Fig. 11); while the results are worse than the full model-based reconstruction, the texture is visually similar. Further, using known models for magnitude may yield better results than using a constant magnitude.

3.2.1 Estimating globally strongest frequencies

The locally strongest frequencies obtained from each patch need to be composed into a single array of frequency locations that will be considered to be the globally strongest frequencies. There may be various combination possibilities for this calculation; for instance, consider the following trade-off: it is not clear a priori whether a frequency with moderate energy, encountered in a large number of patches, is globally strongest than a frequency encountered in a smaller number of patches, but with higher energy in each patch. We choose the following method for this evaluation. A frequency coefficient is assigned a score, defined as follows:

$$s_i = \underline{v}_i^T \cdot \underline{w}_i, \quad (23)$$

where $[\underline{v}_i]_j$ is the number of times a frequency, f_i , has been the j th strongest local frequency, and $\underline{w}_i = [(N)^r, (N-1)^r, \dots, 1^r]$ is a decreasing weights vector for a vector length N with $r = 0.5$. The globally strongest frequencies are the K frequency locations with the highest values of s_i (23). The aforementioned trade-off is not resolved using this proposed method, but it can be controlled by the weighting parameter, r .

4 Reconstruction

One of the main goals in creating image models is reconstruction, that is accomplishing inverse-type tasks such as deblurring and deconvolution. To this end it is necessary to understand how the magnitude and phase coefficients are affected when the image undergoes various degradations.

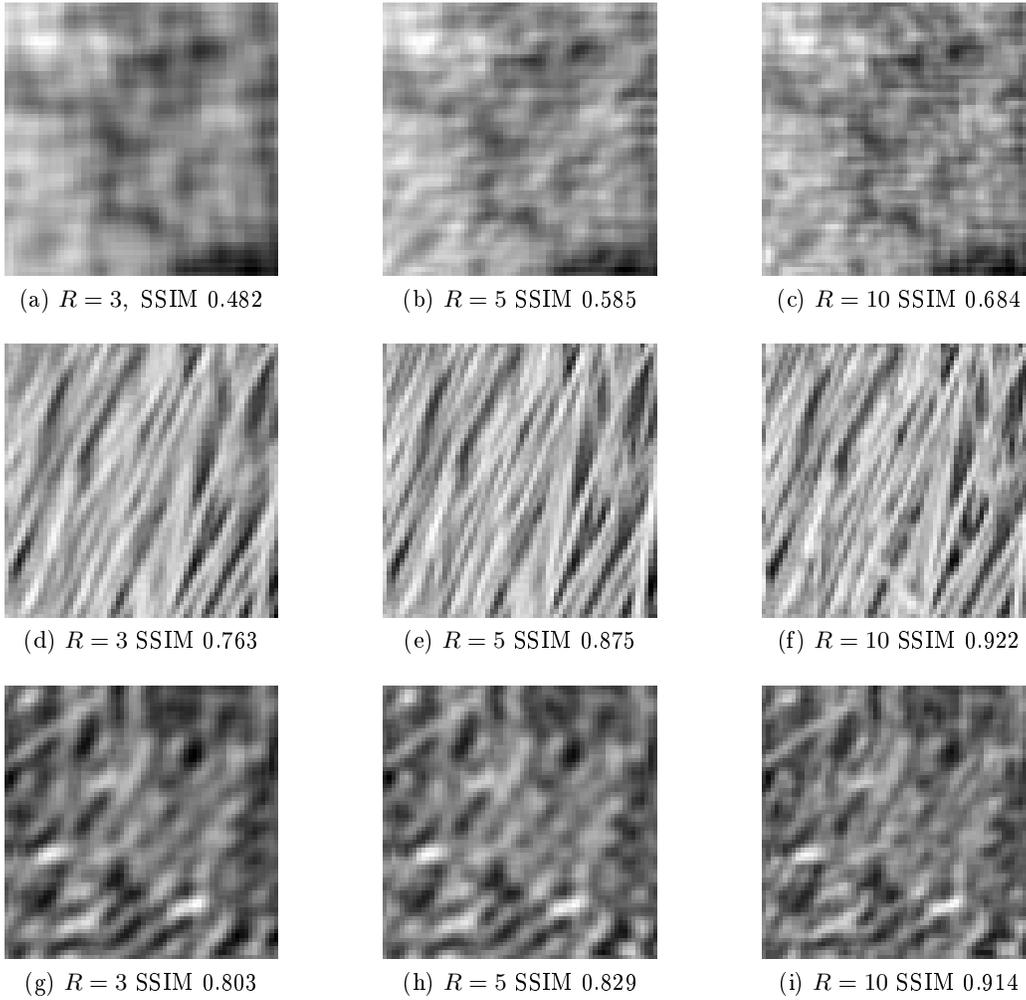


Figure 9: Reconstruction of typical textures with varying number of frequency components (R). Satisfactory reconstruction quality is obtained in anisotropic and isotropic textures (second and third rows, respectively), whereas the isotropic texture (first row) requires a high number of frequency components.

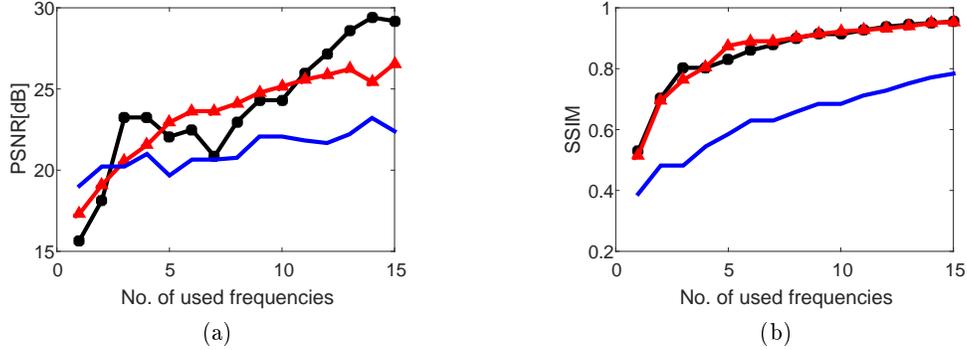


Figure 10: Objective reconstruction assessment as a function of the number of frequency components used (Fig. 9). The isotropic, anisotropic and asotropic images (Fig. 6) are depicted by the blue, red (with triangular markers) and black (with star markers), respectively. The SSIM corresponds better to visual assessment.

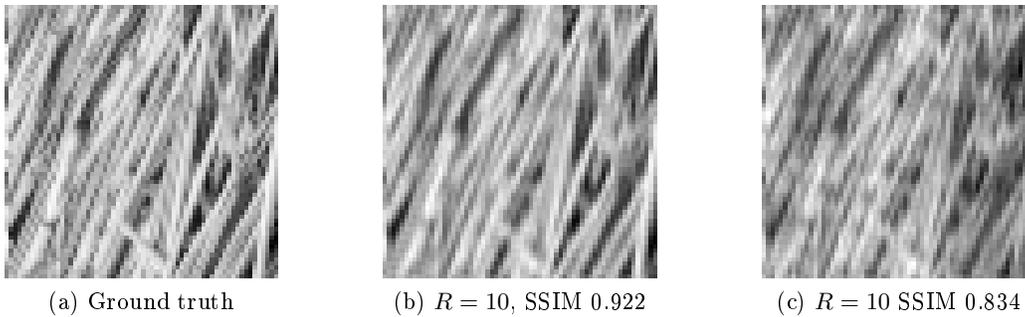


Figure 11: Phase-only reconstruction. for $R = 10$ frequency components. (11b) Image reconstruction by the full model. (11c) Image reconstruction with magnitudes set to 1 in the frequency components used for reconstruction.

4.1 Noise

Similarly to the analysis in Section 3.1, we treat limited support signals. Denoting by $x(t)$ the ground-truth signal and by $n(t)$ the independent Gaussian noise with variance σ^2 , the measurement, $y(t) = x(t) + n(t)$, can be analyzed as follows:

$$\begin{aligned}
 \hat{y}(\omega) &= \int_{-T}^T [x(t) + n(t)] e^{-j\omega t} dt \\
 &= \hat{x}(\omega) + \int_{-T}^T n(t) e^{-j\omega t} dt \\
 &= \hat{x}(\omega) + \int_{-T}^T \cos(\omega t) dt + j \int_{-T}^T \sin(\omega t) dt \\
 &= \hat{x}_r(\omega) + \hat{n}_r(\omega) + j\hat{x}_i(\omega) + j\hat{n}_i(\omega)
 \end{aligned}$$

where $\hat{n}_r(\omega)$ and $\hat{n}_i(\omega)$ are Gaussian with zero mean and the following variances:

$$\begin{aligned}
 \mathbb{E} \left\{ \hat{n}_r(\omega)^2 \right\} &= \sigma^2 \int_{-T}^T \cos(\omega t)^2 dt \\
 &= \sigma^2 \int_{-T}^T \frac{1 + \cos(2\omega t)}{2} dt \\
 &= \sigma^2 \left(T + \int_0^T \cos(2\omega t) dt \right) \\
 &= \sigma^2 \left(T + T \left[\frac{\sin(2\omega t)}{2\omega T} \right]_0^T \right) \\
 &= \sigma^2 T \left(1 + \left[\frac{\sin(2\omega t)}{2\omega T} \right]_0^T \right) = (1 + \text{sinc}(2\omega T)) \sigma^2 T,
 \end{aligned}$$

$$\begin{aligned}
 \mathbb{E} \left\{ \hat{n}_i(\omega)^2 \right\} &= \sigma^2 \int_{-T}^T \sin(\omega t)^2 dt \\
 &= \sigma^2 \int_{-T}^T \frac{1 - \cos(2\omega t)}{2} dt \\
 &= (1 - \text{sinc}(2\omega T)) \sigma^2 T,
 \end{aligned}$$

indicating that the range of variances of $\hat{n}_r(\omega)$ and $\hat{n}_i(\omega)$ is $(0, 2\sigma^2 T)$.

The magnitude of a noisy signal is given by $|\hat{y}(\omega)|^2 = |\hat{x}_r(\omega) + \hat{n}_r(\omega)|^2 + |\hat{x}_i(\omega) + \hat{n}_i(\omega)|^2$. On

average,

$$\begin{aligned}
\mathbb{E} \left\{ |\hat{y}(\omega)|^2 \right\} &= \mathbb{E} \left\{ |\hat{x}_r(\omega) + \hat{n}_r(\omega)|^2 \right\} + \mathbb{E} \left\{ |\hat{x}_i(\omega) + \hat{n}_i(\omega)|^2 \right\} \\
&= \mathbb{E} \left\{ \hat{x}_r^2(\omega) + \hat{n}_r^2(\omega) \right\} + \mathbb{E} \left\{ \hat{x}_i^2(\omega) + \hat{n}_i^2(\omega) \right\} \\
&= \mathbb{E} \left\{ \hat{x}_r^2(\omega) + \hat{x}_i^2(\omega) \right\} + \mathbb{E} \left\{ \hat{n}_r^2(\omega) + \hat{n}_i^2(\omega) \right\} \\
&= \mathbb{E} \left\{ \hat{x}_r^2(\omega) + \hat{x}_i^2(\omega) \right\} + 2\sigma^2 T.
\end{aligned}$$

We observe that on average, the signal’s spectral magnitude is degraded by an addition of a constant value. The relative order of magnitudes is retained, and therefore, estimating the globally strongest frequencies can be performed by using the noisy image directly.

4.2 Blur

A blurred signal, $y(t) = (x * b)(t)$, where $b(t)$ is a blur filter, has a diminished high frequency content. While in many practical cases, the phase is not modified by blurring, the frequency magnitude varies significantly. The order of frequency magnitudes may also change; considering a signal, $x(t)$, and a blur filter, $b(t)$, with the Fourier transforms $\hat{x}(\omega)$ and $\hat{b}(\omega)$, defined as:

$$\begin{aligned}
\hat{x}(\omega) &= \delta(\omega) + 2\delta(\omega \pm 1) + 2\frac{1}{3}\delta(\omega \pm 2) \\
\hat{b}(\omega) &= C \cdot \max \left\{ 0, 1 - \frac{|\omega|}{6} \right\},
\end{aligned}$$

where C is some normalization constant. The blurred image, $y(t)$, has the following Fourier transform:

$$\hat{y}(\omega) = \hat{x}(\omega) \hat{b}(\omega) = C \left[\delta(\omega) + \frac{15}{9}\delta(\omega \pm 1) + \frac{14}{9}\delta(\omega \pm 2) \right],$$

with a changed order of descending magnitudes. Therefore, unlike the case of noisy images, in this case we need to estimate the initial order of magnitudes. In the case of noiseless textures and a blur filter with nonzero frequency response, $\hat{b}(\omega) \neq 0$, we can obtain the original frequencies. In practice, we employ an optimization scheme, as will be described hereinafter.

4.3 Deconvolution of anisotropic textures

The proposed texture model is sparse in the Fourier domain. While many sparsity-based algorithms exploit this property (possibly in other bases than Fourier) [28], our algorithm differs in several aspects. First, we take advantage of the phase of the images, either by using a model for the phase, or by retaining the existing phase; therefore, the processing is performed on the Fourier magnitudes, and not the transform coefficients directly. Further, we assume non-locality in the sense of existence of the globally strong frequencies.

Algorithm 1 A Global magnitude-local phase deconvolution algorithm

1. Partition the image into patches.
 2. Solve the BPDN problem (26) for each patch to estimate the frequency magnitudes.
 3. Obtain a set of K globally-strongest magnitudes via the estimated local magnitudes.
 4. Estimate the local phase for each globally-strongest frequency.
 5. Reconstruct each patch using the globally-strongest magnitudes and estimated phases.
 6. Perform PDE-based post-processing on the entire image.
-

The globally-strongest frequencies are obtained via patch-wise analysis. To obtain the local magnitudes as accurately as possible, the local magnitude estimation of the latter is formulated as follows: let $\hat{a}_+(\omega) \triangleq \hat{a}(\omega)|_{\omega \geq 0}$, i.e. the non-negative frequencies of $\hat{a}(\omega)$, represent the Fourier transform of some signal, $a(t)$. Inasmuch as $a(t)$ is real, knowing $\hat{a}_+(\omega)$ is sufficient for reconstruction of $a(t)$. The same principle applies in 2D images as well, defining $\hat{a}_+(u, v) \triangleq \hat{a}(u, v)|_{u \geq 0}$.

Let $f(x, y)$ and $b(x, y)$ denote a patch and a blur filter and let $\hat{f}(u, v)$ and $\hat{b}(u, v)$ denote their Fourier transforms, respectively. Denote $\hat{y}(u, v)$ as the Fourier transform of the blurred patch, $y(x, y)$, defined in the noiseless case as follows:

$$\begin{aligned}\hat{y}_+(u, v) &= \hat{x}_+(u, v) \hat{b}_+(u, v) \\ |\hat{y}_+(u, v)| &= \left| \hat{b}_+(u, v) \right| |\hat{x}_+(u, v)|\end{aligned}$$

In vectorized form, we have:

$$\underline{\beta} = A\underline{\alpha}, \quad (24)$$

where $\underline{\alpha}$ and $\underline{\beta}$ denote the vectorized versions of $|\hat{x}_+(u, v)|$ and $|\hat{y}_+(u, v)|$, respectively, and $A = \text{diag} \left\{ \left| \hat{b}_+(u, v) \right| \right\}$ is a diagonal matrix. Denoting $\hat{x}_m(u, v)$ to be the model-based representation of $x(u, v)$, we obtain:

$$\hat{x}_m(u, v) = \sum_{j=1}^K \hat{x}_+(u_j, v_j) \delta(u - u_j, v - v_j), \quad (25)$$

where K is the number of coefficients used. Therefore, the vector $\underline{\alpha}$ is sparse. A well-known method to promote sparsity in the solution of (24) is to minimize the L_1 norm. To do so, we use the basis pursuit denoising (BPDN) scheme:

$$\underline{\alpha} = \arg \min_{\underline{\alpha}} \|\underline{\alpha}\|_1 + \lambda \|\underline{\beta} - \hat{a}\underline{\alpha}\|_2. \quad (26)$$

We note that this formulation is commonly used as sparsity-based denoising, but in this case it is used to recover the magnitude vector. Further, it is degenerated, as A is a diagonal matrix.

Using this formulation, we propose the following algorithm for deconvolution of anisotropic textures (Alg. 1). The local phase (step 4) is estimated either via the phase model (22), or by using the phase of the degraded signal, since the phase is, in many cases, invariant to blurring.

This algorithm is proposed for textures that have been severely degraded by combined effects of blur and noise. The dominant non-locality is exploited for reconstruction of oscillating structures. However, the model is too simplistic to fully model complex textures. Therefore, after the texture structure has been recovered, the image undergoes post-processing (step 6) by means of PDE-based deblurring, using anisotropic diffusion [29]. This step completes the reconstruction process, recovering further details.

4.3.1 Remarks

Our formulation of the BPDN scheme (26) is ill-conditioned in the sense that the blur filter yields magnitudes very close to zero, and we cannot expect satisfactory results considering all magnitude coefficients. Therefore, we use the magnitude coefficients of the blur filter to set a threshold; only frequency coefficients in which the blur filter is sufficiently large will be considered for the reconstruction.

While the current method is essentially non-local, the typical image size in our experiments was (64×64 - 128×128), indicating that moderately sized images can be analyzed and processed even in the case of spatially non-homogeneous texture, and possibly images containing texture and other image content. Further, since our reconstruction method is patch-based, a non-homogeneous blur filter can be considered as well, provided it is patch-wise-spatially invariant.

4.4 Example results

We show several typical examples to assess the algorithm’s performance. Our objective assessment will be based on SSIM, that is known to be better than PSNR in the case of textures [30] (both figures are nevertheless calculated and displayed). The leading method in the case of degraded textures shall be a subjective (visual) assessment, as both SSIM and PSNR can yield results that do not correspond to the latter [30].

The algorithm successfully restores details of an 128×128 -sized anisotropic texture (Fig. 12), using patch size of 16×16 and 15 frequency components per patch. We observe that the initial reconstruction (Fig. 12c) has restored much of the ground-truth image structure, even though the image is not strictly anisotropic, in the sense that there are some uncharacteristic directional patterns in some areas of the image. The main observation of the initial reconstruction is that it is almost noise-free, but it appears to be blurred. The post-processed result (Fig. 12f) has been deblurred using PDE-based methods that successfully restored the image structure from the blurred, but noiseless, initial reconstruction. While the reconstruction is not perfect, many of the details have been restored.

While the Wiener filter (Fig. 12d) is considered effective in deconvolution of textures, in the case of severe blur and noise, as in this case, it produces many noisy artifacts. BM3D (Fig. 12e)

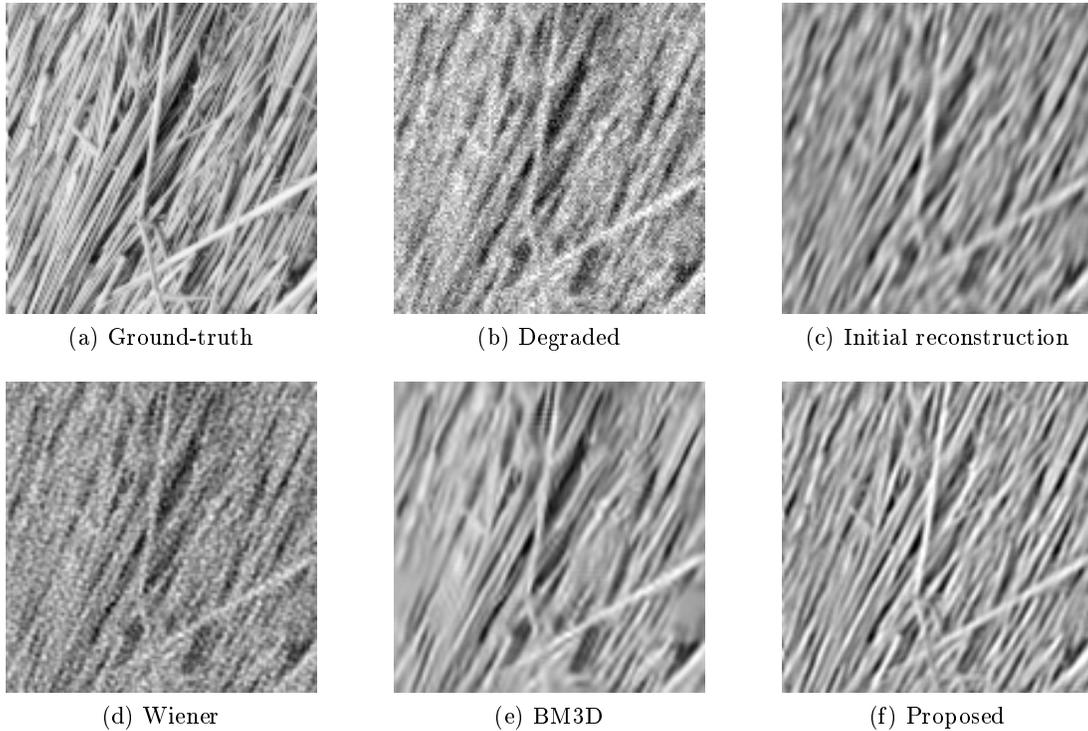


Figure 12: Anisotropic texture deconvolution using Alg. (1). (12a) Ground truth, (12b) image degraded by Gaussian blur with width $\sigma_f = 1$ and AWGN with $\sigma_N = 0.1$ (PSNR 17.47dB, SSIM 0.567), (12c) initial reconstruction (PSNR 18.51dB, SSIM 0.697), (12d) Wiener filtering reconstruction (PSNR 17.37dB, SSIM 0.619), (12e) BM3D result (PSNR 20.43dB, SSIM 0.713), and (12f) proposed algorithm result (PSNR 20.33dB, SSIM 0.841).

reconstructs some of the degraded edges, but much of the detail is still missing in the reconstructed image.

The second example is of 128×128 -sized isotropic textures (Fig. 13 and Fig. 14). Similarly to the previous example, and using the same degradation and reconstruction parameters, we observe the recovered details in the initial reconstruction (Fig. 13c and Fig. 14c, respectively), and the deblurred and recovered details of the final reconstruction (Fig. 13f and Fig. 14f, respectively). This texture’s orientation is less pronounced than the previous example, but details are still reconstructed successfully, due to the fact that the same characteristic frequencies are dominant throughout the image.

To further assess the algorithm, the algorithm was evaluated on 10 randomly selected images from the Brodatz database [31] (Table 1). The degradation was performed by Gaussian and motion blurring of various widths and lengths, respectively, and two values of noise variances. The results indicate that in the case of textures, the proposed algorithm produces better SSIM scores on almost all experiments, corresponding to the subjective results presented in the previous experiments on

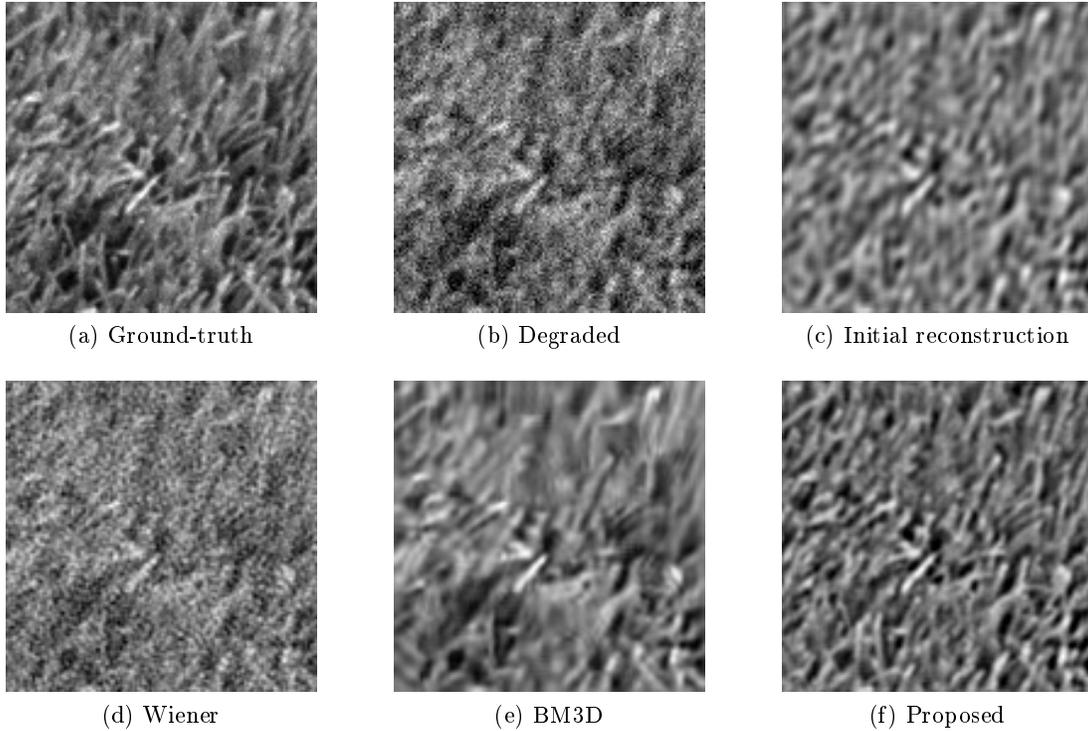


Figure 13: Asotropic texture deconvolution using Alg. (1). (13a) Ground truth, (13b) image degraded by Gaussian blur with width $\sigma_f = 1$ and AWGN with $\sigma_N = 0.1$ (PSNR 19.24dB, SSIM 0.549), (13c) initial reconstruction (PSNR 19.41dB, SSIM 0.710), (13d) Wiener filtering reconstruction (PSNR 19.5dB, SSIM 0.605), (13e) BM3D result (PSNR 22.38dB, SSIM 0.755), and (13f) proposed algorithm result (PSNR 22.51dB, SSIM 0.879).

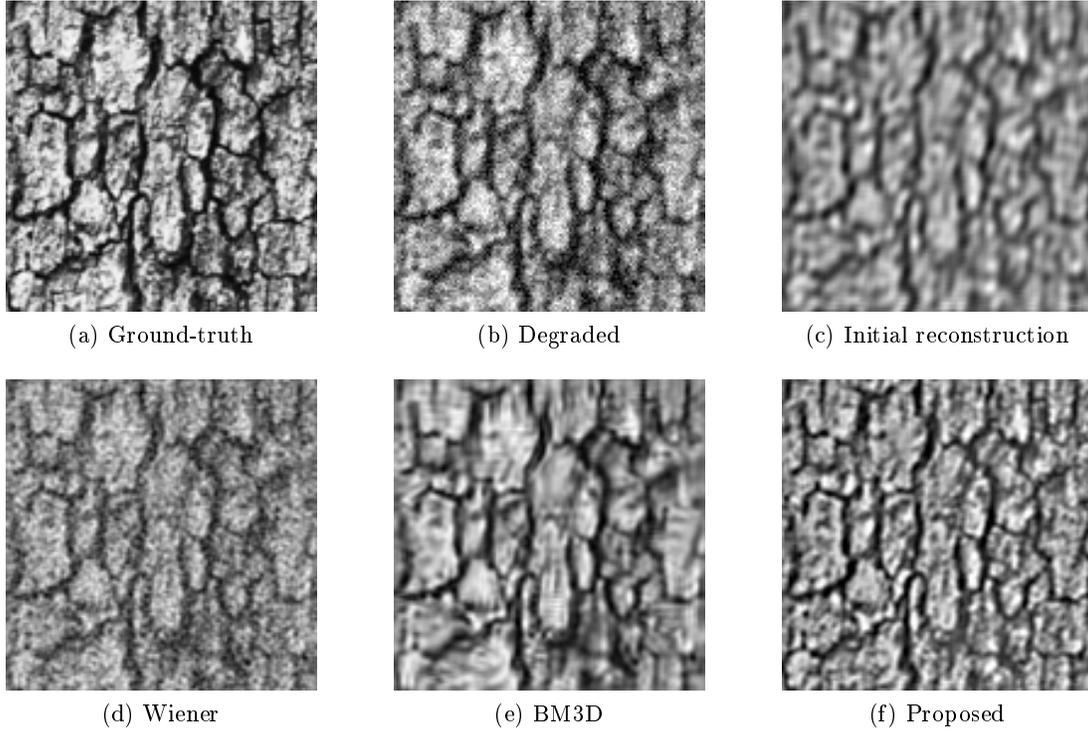


Figure 14: A second anisotropic texture deconvolution, using the same degradation and reconstruction parameters as Fig. 13). (14a) Ground truth, (14b) degraded image (PSNR 17.48dB, SSIM 0.692), (14c) initial reconstruction (PSNR 16.39dB, SSIM 0.618), (14d) Wiener filtering reconstruction (PSNR 17.49dB, SSIM 0.696), (14e) BM3D result (PSNR 19.23dB, SSIM 0.760), and (14f) proposed algorithm result (PSNR 19.75dB, SSIM 0.897).

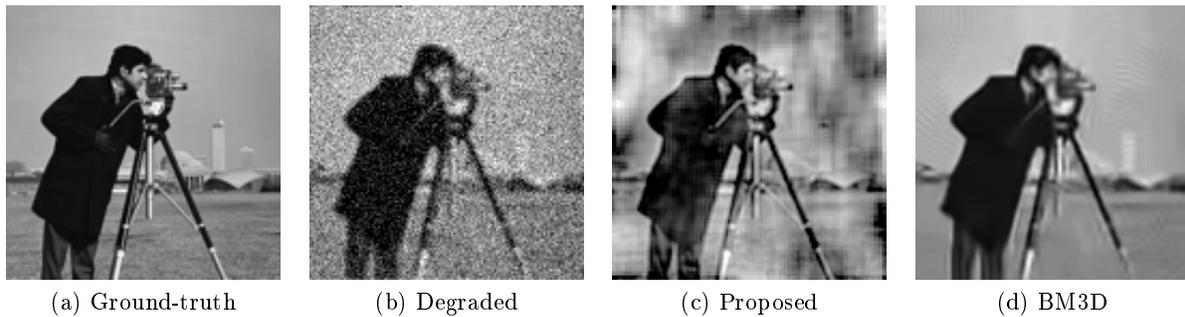


Figure 15: Non-texture deconvolution. (15a) Ground truth, (15b) image degraded by Gaussian blur with width $\sigma_f = 1$ and AWGN with $\sigma_N = 0.1$ (PSNR 19.51dB, SSIM 0.317), (15c) proposed algorithm (PSNR 16.49dB, SSIM 0.553), and (15d) BM3D result (PSNR 24.92, SSIM 0.804).

individual images (Fig. 12, Fig. 13 and Fig. 14).

In the case of non-textured, non-homogeneous content, the algorithm does not deconvolve the images properly (Fig. 15). This is due to the fact that the non-local assumption means attempting to perform local reconstruction using unsuitable frequency coefficients. In this work we assume, however, that images of a suitable type can be correctly classified prior to using the algorithm.

5 A combined model for isotropic and anisotropic textures

In Section 3 we analyzed the phase of anisotropic textures. In previous studies [7, 32] we have addressed the isotropic, fractal textures, and proposed the fractional Brownian motion (fBm) [18] as a suitable model. In this section we focus on modeling textures containing both elements, structured as well as stochastic.

5.1 Definition of the model

Recall Eq. (25):

$$\hat{x}_m(u, v) = \sum_{j=1}^K \hat{x}_+(u, v) \delta(u - u_j, v - v_j),$$

the anisotropic texture model, defined on a half-plane of the frequency domain. Let $\hat{x}(u, v)$ denote the frequency response on the entire plane. According to the model, we can consider the frequency response to be a sum of finite frequency coefficients. Consider a single coefficient pair of symmetric frequency components in 1D, denoting the frequency coefficients and the frequency by $a_i \in \mathbb{C}$ and f_i , respectively, then

$$\begin{aligned} |a_i| e^{j\angle a_i} \delta(f - f_i) + |a_i| e^{-j\angle a_i} \delta(f + f_i) &\stackrel{\mathcal{F}}{\leftrightarrow} |a_i| \left(e^{j\angle a_i} e^{j\omega_i t} + e^{-j\angle a_i} e^{-j\omega_i t} \right) \\ &= |a_i| \left(e^{j(\angle a_i + \omega_i t)} + e^{-j(\angle a_i + \omega_i t)} \right) \\ &= 2 |a_i| \cos(\omega_i t + \angle a_i). \end{aligned} \tag{27}$$

	G $\sigma_G = 0.5$	G $\sigma_G = 1$	G $\sigma_G = 2$	M $S = 3$	M $S = 5$	M $S = 7$
Noisy	24.31[dB]	20.06[dB]	18.01[dB]	21.68[dB]	19.55[dB]	18.10[dB]
Wiener	21.47[dB]	20.01[dB]	18.55[dB]	19.13[dB]	17.88[dB]	17.24[dB]
BM3D	25.73[dB]	22.20[dB]	20.32[dB]	22.91[dB]	21.48[dB]	20.23[dB]
Our	27.32[dB]	22.50[dB]	19.49[dB]	24.20[dB]	21.09[dB]	18.99[dB]

(a) PSNR[dB], noise standard deviation $\sigma = 0.05$

	G $\sigma_G = 0.5$	G $\sigma_G = 1$	G $\sigma_G = 2$	M $S = 3$	M $S = 5$	M $S = 7$
Noisy	19.68[dB]	17.77[dB]	16.66[dB]	18.56[dB]	17.50[dB]	16.66[dB]
Wiener	17.92[dB]	18.31[dB]	17.42[dB]	17.26[dB]	16.53[dB]	16.00[dB]
BM3D	22.48[dB]	20.79[dB]	19.37[dB]	21.34[dB]	20.10[dB]	19.00[dB]
Our	25.01[dB]	22.32[dB]	19.83[dB]	23.17[dB]	20.90[dB]	19.12[dB]

(b) PSNR[dB], noise standard deviation $\sigma = 0.1$

	G $\sigma_G = 0.5$	G $\sigma_G = 1$	G $\sigma_G = 2$	M $S = 3$	M $S = 5$	M $S = 7$
Noisy	0.832	0.637	0.486	0.724	0.597	0.485
Wiener	0.762	0.669	0.588	0.675	0.582	0.523
BM3D	0.870	0.749	0.655	0.783	0.705	0.638
Our	0.936	0.824	0.660	0.864	0.730	0.586

(c) SSIM, noise standard deviation $\sigma = 0.05$

	G $\sigma_G = 0.5$	G $\sigma_G = 1$	G $\sigma_G = 2$	M $S = 3$	M $S = 5$	M $S = 7$
Noisy	0.645	0.477	0.355	0.553	0.448	0.359
Wiener	0.571	0.559	0.501	0.537	0.467	0.411
BM3D	0.769	0.670	0.585	0.708	0.641	0.570
Our	0.898	0.799	0.635	0.832	0.704	0.564

(d) SSIM, noise standard deviation $\sigma = 0.1$

Table 1: Deconvolution results. (1a), (1b): The PSNR results for $\sigma = 0.05$ and $\sigma = 0.1$, respectively. (1c), (1d): The SSIM results for $\sigma = 0.05$ and $\sigma = 0.1$, respectively. Columns indicated by G represent a Gaussian filter with standard deviation σ_G , specified next to each result, and columns indicated by M represent a 45° motion blur with size S , specified next to each result, respectively. The results in all tables are averaged over all images in the dataset. The maximal results, shown in bold-faced font, were obtained by our algorithm in most of the scenarios.

A similar analysis can be performed for the 2D case:

$$\begin{aligned}
& |a_i| e^{j\angle a_i} (\delta(u - u_i) \delta(v - v_i) + \delta(u + u_i) \delta(v + v_i)) \\
& \stackrel{\mathcal{F}_X}{\Leftrightarrow} |a_i| e^{j\angle a_i} (e^{-ju_i x} \delta(v - v_i) + e^{+ju_i x} \delta(v + v_i)) \\
& \stackrel{\mathcal{F}_Y}{\Leftrightarrow} |a_i| e^{j\angle a_i} (e^{-ju_i x} e^{-jv_i y} + e^{+ju_i x} e^{+jv_i y}) \\
& |a_i| e^{j\angle a_i} (e^{-j(u_i x + v_i y)} + e^{+j(u_i x + v_i y)}) \\
& = |a_i| (e^{-j(u_i x + v_i y + \angle a_i)} + e^{+j(u_i x + v_i y + \angle a_i)}) \\
& = 2 |a_i| \cos(u_i x + v_i y + \angle a_i), \tag{28}
\end{aligned}$$

where \mathcal{F}_I denotes Fourier transform in the $I \in \{X, Y\}$ direction. Therefore, the model can be represented in the spatial domain as follows:

$$c(x, y) = \sum_{i=1}^K m_i \cos(u_i x + v_i y + \theta_i), \tag{29}$$

where $\{m_i, \theta_i\}_{i=1}^K$ are the magnitude-phase pair and $\{u_i, v_i\}_{i=1}^K$ are the frequency locations.

Let $c(x, y)$ be an anisotropic texture (29), and let $f(x, y)$ be a 2D-fGn/fBm image with Hurst parameter $H \in (0, 1)$. Assume H , $\{m_i\}_{i=1}^K$ and $\{u_i, v_i\}_{i=1}^K$ are global parameters (applicable across patches), and $\{\theta_i\}_{i=1}^K$ are local parameters. Denote the combined model by $g(x, y)$:

$$g(x, y) = \alpha f(x, y) + (1 - \alpha) c(x, y), \tag{30}$$

where $\alpha \in [0, 1]$ is a weighting factor. The images $f(x, y)$ and $c(x, y)$ are assumed to be independent.

We note that Wold's theorem [26] appears to be applicable to the decomposition of $g(x, y)$ (30), but this is not the case. In Wold's theorem, a wide-sense stationary process is separated into a predictable process and a stationary process, where the predictable process has a punctate spectrum (sum of Dirac delta functions). This requires the autocorrelation to be composed of sines and cosines. In our case, however, the space-domain signal has a constant phase, rendering it to be non-stationary, since in a predictable signal, the phase should be uniformly distributed. Further, if $f(x, y)$ is assumed to be fBm, the process itself is not stationary.

In our formulation hereinafter we assume that $f(x, y)$ is either fGn or fBm, but the derivation can be applied to any zero-mean Gaussian image with a known covariance matrix.

(PAPER ONLY:) An optimal estimator in the MMSE sense can be derived for the mixture parameter, α , given a texture image, $g(x, y)$. This is derived elsewhere using statistics of natural textures [33].

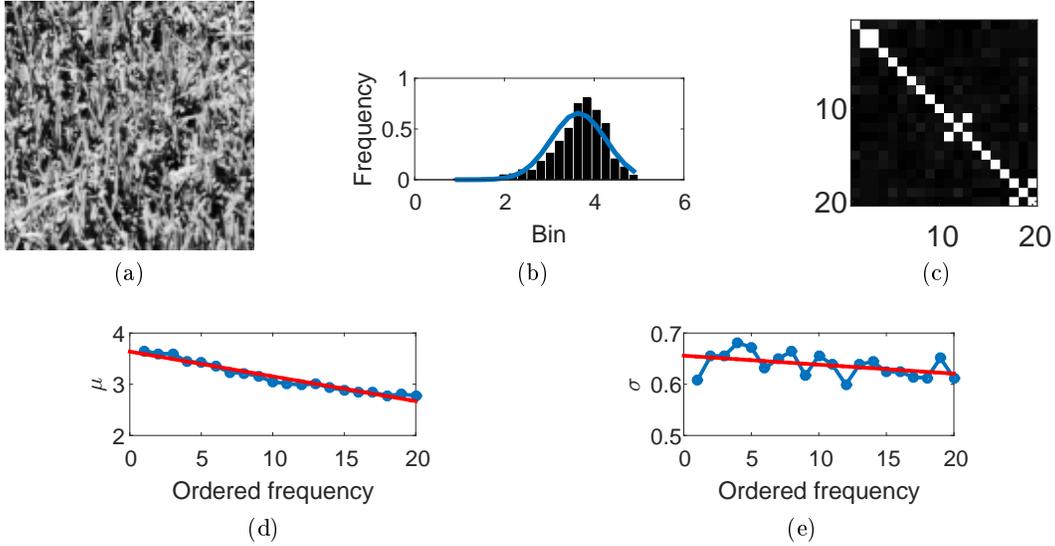


Figure 16: Example of the coefficient model with 10 strongest-magnitude frequency coefficients. (16a) Image from Brodatz, (16b) strongest frequency coefficient log-histogram across all patches (with fitted Gaussian pdf overlaid in blue), (16c) correlation matrix of the dominant frequencies (bright values correspond to high absolute correlation), (16d) and (16e) coefficient’s mean and standard deviation, respectively; the samples are depicted in blue with round markers and the linear fit is depicted in red. This texture approximately satisfies the model assumptions.

5.1.1 Statistics of frequency coefficient, \underline{c}

We propose the following model for \underline{c} , the magnitude coefficients: each of the non-zero elements of $\tilde{\underline{c}} = \log(\underline{c})$ is independent of the other elements. Let $\tilde{c}_{[j]}$ be the j th largest element in $\tilde{\underline{c}}$. Its variance and its mean decline linearly with j :

$$\begin{aligned} \mathbb{E} \{ \tilde{c}_{[j]} \} &= p_0 + p_1 j, & p_1 < 0. \\ \sigma_{\tilde{c}_{[j]}} &= q_0 + q_1 j, & q_1 < 0. \end{aligned}$$

The variables that should be estimated prior to estimation of α are, therefore, p_0 , p_1 , q_0 and q_1 . This is an approximated model, based on analysis of the Brodatz dataset [31]. While its accuracy is not statistically significant (Fig. 16 and Fig. 17), it will suffice for our application. We note that some textures do not satisfy the assumptions in full (Fig. 18).

The global parameters that are also required are the phases, θ , and the dominant frequency locations of \underline{c} .

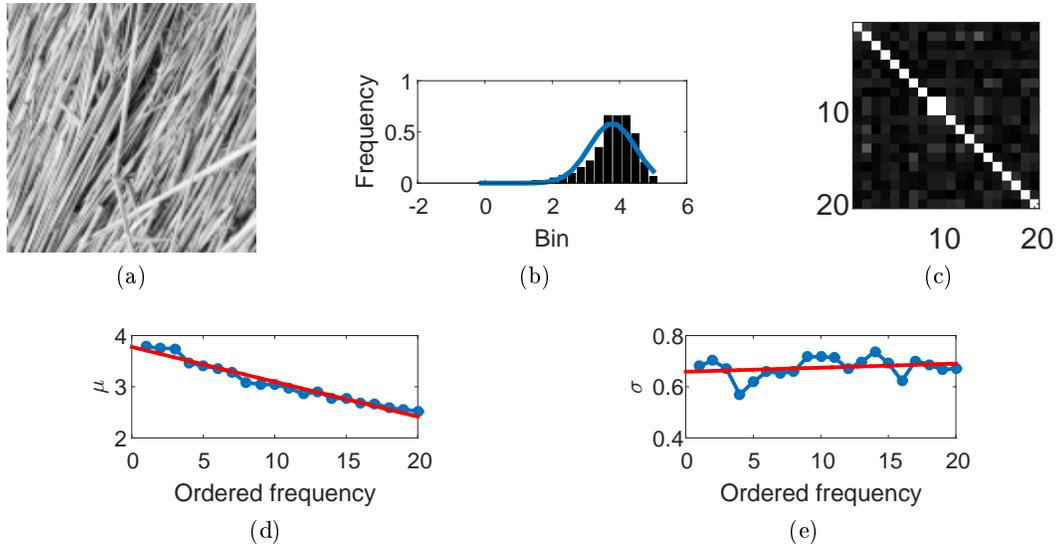


Figure 17: A similar example for a Brodatz texture, approximately satisfying the model assumptions. The figure definitions are similar to Fig. 16.

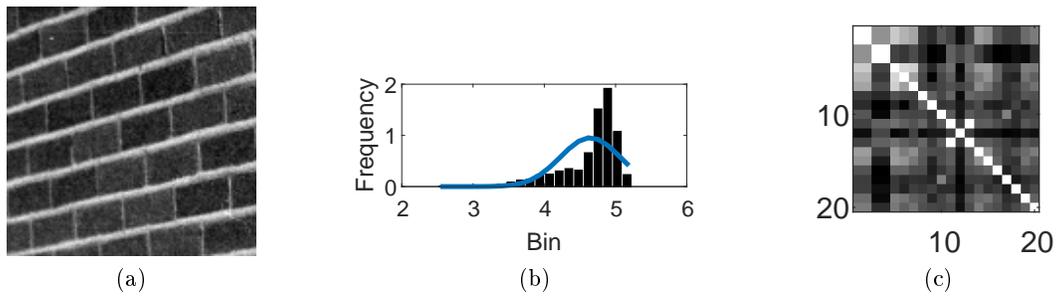


Figure 18: A Brodatz texture not satisfying the model assumptions, as indicated by the non-Gaussian coefficient histogram (18b), and the dependency between coefficients (18c).

5.2 Estimation of α for a 2D signal

We are interested in quantifying the fractal content of a textured image. This can be indicated by estimation of the parameter α (30), which corresponds to the ratio of the fGn/fBm content in the image, relative to the anisotropic content.

In this section we use following vectorized (column-stacked) formulation for 2D images:

$$\underline{g} = \alpha \underline{f} + (1 - \alpha) \underline{c} = \alpha \cdot \Sigma_H^{0.5} \underline{w} + (1 - \alpha) \cdot T_\theta \underline{m} \quad (31)$$

$$\triangleq \alpha \underline{S}_1 + (1 - \alpha) \underline{S}_2, \quad (32)$$

where \underline{w} is a standard i.i.d normal vector, Σ_H is a the covariance for 2D-fGn/fBm with some $H \in (0, 1)$, \underline{c} is a log-normal random vector; T_θ is a 2D operator matrix (operating on vectorized images), e.g.

$$T_\theta = D_\theta^1 \otimes D_\theta^1, \quad T_\theta = F_\theta^1 \otimes F_\theta^1,$$

where \otimes denotes the Kronecker product, D_θ^1 is a 1D-DCT matrix and F_θ^1 is a 1D-DFT matrix, in which all harmonics are shifted by a known phase, θ . In general, T_θ can be any transform in which the phase is controllable. A shift in phase does not change the unitary property of such transforms (Appendix B). We note that an additional representation will be $T_\theta^1 C (T_\theta^1)^T$ where C is the image matrix corresponding to \underline{c} . The phase, θ , is assumed to be known and is chosen for each patch according to the phase model (estimated globally). The frequency locations, $\{i : c_i > 0\}$, are determined globally as well.

The anisotropic texture is sparse in the transform domain, i.e. $|\underline{c}|_0 = K \ll N$ where N is the number of elements in \underline{c} . To include the sparsity in the model (31), let $U \in \{0, 1\}^{n \times K}$ denote a rectangular indicator matrix that creates \underline{c} from \underline{c}' : $\underline{c} \triangleq U \underline{c}'$, where $\underline{c}' \in \mathbb{R}^K$ contains the nonzero values of \underline{c} .

The mixture parameter, α , is assumed to be random:

$$\alpha \sim \mathcal{U}[0, 1]$$

$$\mathbb{E}\{\alpha\} = 0.5$$

$$\mathbb{E}\{\alpha^2\} = \mathbb{E}\{(1 - \alpha)^2\} = \frac{1}{3}.$$

We consider the linear MMSE estimator:

$$\hat{\alpha}_{\text{lin}}(\underline{g}) = \mathbb{E}\{\alpha\} + \text{Cov}\{\alpha, \underline{g}\}^T \Lambda_g^{-1} (\underline{g} - \mathbb{E}\{\underline{g}\}),$$

where

$$\begin{aligned}
\underline{\mu}_{s_2} &\triangleq \mathbb{E} \{ \underline{S}_2 \} \\
\underline{m} &\triangleq \mathbb{E} \{ (1 - \alpha) \underline{S}_2 \} = \frac{1}{2} \mathbb{E} \{ \underline{S}_2 \} = \frac{1}{2} \underline{\mu}_{s_2} \\
\mathbb{E} \{ \underline{g} \} &= \mathbb{E} \{ \alpha \underline{S}_1 + (1 - \alpha) \underline{S}_2 \} = \mathbb{E} \{ \alpha \} \mathbb{E} \{ \underline{S}_1 \} + \underline{m} = \frac{1}{2} \underline{\mu}_{s_2}.
\end{aligned}$$

The last equality follows from the zero-mean assumption of the fBm/fGn. The covariance of \underline{g} is given by:

$$\begin{aligned}
\Lambda_g &= \mathbb{E} \left\{ (\alpha \underline{S}_1 + (1 - \alpha) \underline{S}_2 - \underline{m}) (\alpha \underline{S}_1 + (1 - \alpha) \underline{S}_2 - \underline{m})^T \right\} \\
&= \frac{1}{3} \mathbb{E} \{ \underline{S}_1 \underline{S}_1^T \} + \mathbb{E} \left\{ ((1 - \alpha) \underline{S}_2 - \underline{m}) ((1 - \alpha) \underline{S}_2 - \underline{m})^T \right\} \\
&= \frac{1}{3} \Sigma_H + \Lambda_2,
\end{aligned}$$

where

$$\begin{aligned}
\Lambda_2 &= \mathbb{E} \left\{ ((1 - \alpha) \underline{S}_2 - \underline{m}) ((1 - \alpha) \underline{S}_2 - \underline{m})^T \right\} \\
&= \mathbb{E} \left\{ (1 - \alpha)^2 \underline{S}_2 \underline{S}_2^T - (1 - \alpha) \underline{m} \underline{S}_2^T - (1 - \alpha) \underline{S}_2 \underline{m}^T + \underline{m} \underline{m}^T \right\} \\
&= \frac{1}{3} \mathbb{E} \{ \underline{S}_2 \underline{S}_2^T \} - \frac{1}{2} \underline{m} \mathbb{E} \{ \underline{S}_2 \}^T - \frac{1}{2} \mathbb{E} \{ \underline{S}_2 \} \underline{m}^T + \underline{m} \underline{m}^T \\
&= \frac{1}{3} \mathbb{E} \{ \underline{S}_2 \underline{S}_2^T \} - \frac{1}{2} \cdot \frac{1}{2} \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T - \frac{1}{2} \cdot \frac{1}{2} \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T + \frac{1}{4} \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T \\
&= \frac{1}{3} \left(\Lambda_{S_2} + \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T \right) - \frac{1}{4} \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T \\
&= \frac{1}{3} \Lambda_{S_2} + \frac{1}{12} \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T.
\end{aligned}$$

Further the covariance between α and \underline{g} is given by:

$$\begin{aligned}
\text{Cov}\{\alpha, \underline{g}\} &= \mathbb{E}\{(\alpha - \mathbb{E}\{\alpha\})(\underline{g} - \mathbb{E}\{\underline{g}\})\} \\
&= \mathbb{E}\left\{\left(\alpha - \frac{1}{2}\right)\left(\underline{g} - \frac{1}{2}\underline{\mu}_{s_2}\right)\right\} \\
&= \mathbb{E}\{\alpha\underline{g}\} - \frac{1}{2}\mathbb{E}\{\underline{g}\} - \frac{1}{2}\underline{\mu}_{s_2}\mathbb{E}\{\alpha\} + \frac{1}{4}\underline{\mu}_{s_2} \\
&= \mathbb{E}\{\alpha\underline{g}\} - \frac{1}{4}\underline{\mu}_{s_2} \\
&= \mathbb{E}\{\alpha(\alpha\underline{S}_1 + (1-\alpha)\underline{S}_2)\} - \frac{1}{4}\underline{\mu}_{s_2} \\
&= \mathbb{E}\{\alpha^2\underline{S}_1 + \alpha(1-\alpha)\underline{S}_2\} - \frac{1}{4}\underline{\mu}_{s_2} \\
&= \frac{1}{3}\mathbb{E}\{\underline{S}_1\} + \mathbb{E}\{\alpha - \alpha^2\}\mathbb{E}\{\underline{S}_2\} - \frac{1}{4}\underline{\mu}_{s_2} \\
&= \underline{\mu}_{s_2}\left(\frac{4}{24} - \frac{6}{24}\right) = -\frac{1}{12}\underline{\mu}_{s_2}.
\end{aligned}$$

The linear estimator is, therefore, given by:

$$\hat{\alpha}_{\text{lin}}(\underline{g}) = \text{Cov}\{\alpha, \underline{g}\}^T \Lambda_g^{-1} (\underline{g} - \mathbb{E}\{\underline{g}\}) \quad (33)$$

$$= \frac{1}{2} - \frac{1}{12}\underline{\mu}_{s_2}^T \left(\frac{1}{3}\Sigma_H + \frac{1}{3}\Lambda_{S_2} + \frac{1}{12}\underline{\mu}_{s_2}\underline{\mu}_{s_2}^T\right)^{-1} \left(\underline{g} - \frac{1}{2}\underline{\mu}_{s_2}\right) \quad (34)$$

$$= \frac{1}{2} - \underline{\mu}_{s_2}^T \left(4\Sigma_H + 4\Lambda_{S_2} + \underline{\mu}_{s_2}\underline{\mu}_{s_2}^T\right)^{-1} \left(\underline{g} - \frac{1}{2}\underline{\mu}_{s_2}\right). \quad (35)$$

This estimator is a function of the first- and second-order statistics of \underline{S}_2 , $\underline{\mu}_{S_2}$ and Λ_{S_2} :

$$\underline{S}_2 = T_\theta \underline{c} = T_\theta U \underline{c}' \quad (36)$$

$$\underline{\mu}_{S_2} = T_\theta U \mathbb{E}\{\underline{c}\} \quad (37)$$

$$\text{Cov}\{\underline{S}_2\} = \text{Cov}\{T_\theta U \underline{c}'\} = T_\theta U \text{Cov}\{\underline{c}'\} U T_\theta^T. \quad (38)$$

The determination of the statistics of \underline{c} , the magnitude coefficients of the anisotropic texture, is performed empirically.

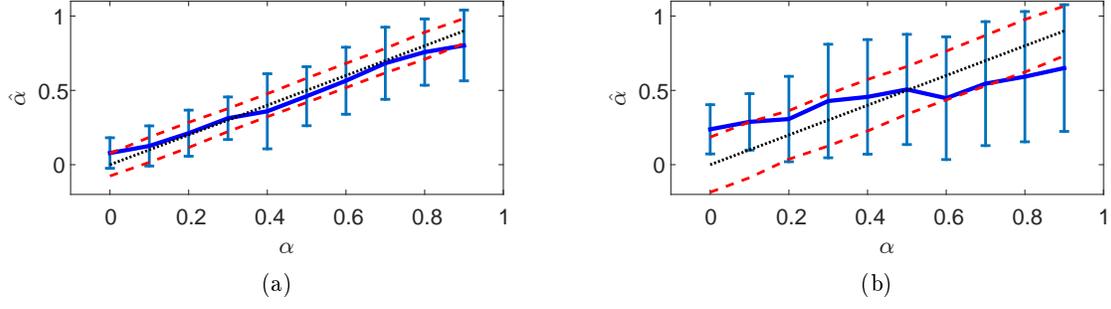


Figure 19: Simulation of $\hat{\alpha}_{lin}(\underline{g})$ for two values of K and patch size of 16×16 : (19a) $K = 50$, (19b) $K = 10$. Each point depicts the averaged result of 200 evaluations. The true values of α and the estimated values are shown in dashed-black and blue, respectively. The theoretic estimator standard deviations (41) are shown in dashed-red, and the error bars indicate the empirical standard deviation of the estimator.

5.2.1 Performance of the estimator $\hat{\alpha}_{lin}(\underline{g})$

The estimator error, $\mathbb{E} \left\{ (\alpha - \hat{\alpha}_{lin}(\underline{g}))^2 \right\}$, for a linear MMSE estimator, is given by [26]:

$$\mathbb{E} \left\{ (\alpha - \hat{\alpha}_{lin}(\underline{g}))^2 \right\} = \frac{1}{12} - \text{Cov}\{\alpha, \underline{g}\}^T \text{Cov}\{\underline{g}\}^{-1} \text{Cov}\{\alpha, \underline{g}\} \quad (39)$$

$$= \frac{1}{12} - \left(-\frac{1}{12} \underline{\mu}_{s_2}^T \right) \left(\frac{1}{3} \Sigma_H + \frac{1}{3} \Lambda_{S_2} + \frac{1}{12} \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T \right)^{-1} \left(-\frac{1}{12} \underline{\mu}_{s_2} \right) \quad (40)$$

$$= \frac{1}{12} \left[1 - \underline{\mu}_{s_2}^T \left(4 \Sigma_H + 4 \Lambda_{S_2} + \underline{\mu}_{s_2} \underline{\mu}_{s_2}^T \right)^{-1} \underline{\mu}_{s_2} \right]. \quad (41)$$

Performance assessment of the estimator is obtained by means of simulation, using $T_\theta^1 = D_\theta^1$ where D_θ^1 is a 1D DCT matrix. A set of signals are simulated according to the model (21), using the following parameters: patch size of 16×16 , Hurst parameter $H = 0.5$ and θ chosen arbitrarily to have the value of 0.5. The simulated values of α span the range $[0, 0.9]$ in step-size of 0.1. The case of $\alpha = 1$ is not simulated, as it misses the structural part completely. The simulated results for $K = \{50, 10\}$ (Fig. 19) indicate that the estimator performs better in the non-sparse case ($K = 50$), due to the increased energy of the structural part of the signal. However, the practical case is the latter, i.e. $K = 10$. We note that in both cases, the estimator performs reasonably within the range of the theoretical standard errors. The relatively high empirical standard deviation of the estimator accounts for both the estimator error, and the errors due to the covariance and mean estimation (38), performed empirically from the simulated data.

5.3 Finding a suitable representation for structured textures

We investigate several approaches to the modelling of structured textures; our method is patch-based and we, therefore, formulate the structured part as a vector, which shall be a linear operator, T , applied on a vector of coefficients, \underline{c} . Our texture model assumes punctate spectra of symmetric Diracs, each having the following spatial representation (28):

$$\begin{aligned} & |a_i| e^{j\angle a_i} (\delta(u - u_i) \delta(v - v_i) + \delta(u + u_i) \delta(v + v_i)) \\ & \stackrel{\mathcal{F}}{\Leftrightarrow} 2 |a_i| \cos(u_i x + v_i y + \angle a_i), \end{aligned}$$

where \mathcal{F} denotes the 2D Fourier transform. While this can be approximated by DCT or similar separable transforms, An accurate, suitable, transform should have basis elements in the form of $\cos(u_i x + v_i y + \angle a_i)$, rendering it to be non-separable - a disadvantage with respect to the ordinary Fourier transform (or its discrete counterpart, the DFT or DCT).

5.3.1 A real-valued transform approach

Consider the following non-separable, real-valued transform:

$$[T]_{u,v} = \cos(ux + vy + \angle a_i).$$

It can be verified that this transform is neither separable nor orthogonal (Appendix (C)). Thus, while providing a sparse representation for a texture satisfying the model assumptions, discarding low-energy coefficients will yield undesired effects in the resulting image. Despite the relative simplicity of the real-valued transform, we will employ the DFT.

5.3.2 A DFT-based, sparsity enhancing, optimization scheme

Using the DFT matrix, F_θ , as the linear operator applied on a set of complex coefficients, \underline{c} , our goal is to find a sparse magnitude vector. Using the fact that \underline{c} is a sparse vector (31), we propose the following scheme for finding \underline{c} given θ and H , using the fact that the likelihood of \underline{g} given \underline{c} is

a Gaussian, and assuming a Laplacian (L_1 -based) distribution for \underline{c} to reward sparsity:

$$\begin{aligned}
\hat{\underline{c}} &= \arg \max_{\underline{c}} p(\underline{c}|g) \\
&= \arg \max_{\underline{c}} p(g|\underline{c}) p(\underline{c}) \\
&= \arg \max_{\underline{c}} \exp \left\{ -\frac{1}{2} (\underline{g} - (1 - \alpha) F_{\theta} \underline{c})^T \frac{1}{\alpha^2} \mathbf{R}_H (\underline{g} - (1 - \alpha) F_{\theta} \underline{c}) \right\} e^{-\frac{\lambda_1}{2} \|\underline{c}\|_1} \\
&= \arg \min_{\underline{c}} (\underline{g} - (1 - \alpha) F_{\theta} \underline{c})^T \frac{1}{\alpha^2} \mathbf{R}_H (\underline{g} - (1 - \alpha) F_{\theta} \underline{c}) + \lambda_1 \|\underline{c}\|_1 \\
&= \arg \min_{\underline{c}} \left(\frac{1}{\alpha} \Phi \underline{g} - \frac{1}{\alpha} \Phi (1 - \alpha) F_{\theta} \underline{c} \right)^T \left(\frac{1}{\alpha} \Phi \underline{g} - \frac{1}{\alpha} \Phi (1 - \alpha) F_{\theta} \underline{c} \right) + \lambda_1 \|\underline{c}\|_1 \\
&= \arg \min_{\underline{c}} \left\| \frac{1}{\alpha} \Phi \underline{g} - \frac{1}{\alpha} \Phi (1 - \alpha) F_{\theta} \underline{c} \right\|_2^2 + \lambda_1 \|\underline{c}\|_1
\end{aligned} \tag{42}$$

where $\mathbf{R}_H = \Phi \Phi^T$ is the fBm/fGn covariance matrix (4). Denoting an R -norm to be

$$\|\cdot\|_R^2 \triangleq |\cdot|^T R^{-1} |\cdot|, \tag{43}$$

the problem can be formulated for $\underline{c} \in \mathbb{R}^K$ as follows:

$$\hat{\underline{c}} = \arg \min_{\underline{c}} \left\| \frac{1}{\alpha} \underline{g} - \frac{1}{\alpha} (1 - \alpha) F_{\theta}^{-1} \underline{c} \right\|_R^2 + \lambda \|\underline{c}\|_1 \tag{44}$$

where \underline{g} is the given patch, F_{θ}^{-1} is a phase-shifted inverse DFT matrix, $\underline{c} = |\underline{c}| \exp\{j\angle \underline{c}\}$, and the norms are complex. We note that this problem differs from the usual Lasso problem [28] in that \underline{c} is a complex vector. We shall now obtain an iterative solution of (44).

5.3.3 A preliminary step: the unitary case

Consider the following model:

$$\underline{g} = F^{-1} \underline{c} + \underline{n},$$

where F is an inverse DFT matrix, \underline{c} are complex coefficients, and $\underline{n} \sim \mathcal{N}(0, \sigma^2 I)$ is an independent noise source. We wish to obtain a sparse solution in the magnitude of the vector \underline{c} :

$$\hat{\underline{c}} = \arg \min_{\underline{c}} L := \|\underline{g} - F^{-1} \underline{c}\|^2 + \lambda \|\underline{c}\|_1.$$

We observe that the L_1 norm is non-differentiable with a complex vector \underline{c} :

$$\frac{\partial (\|\underline{c}\|_1)}{\partial c_i} = \frac{\partial |c^i|}{\partial c_i}.$$

Considering the function $f(z) = u(a, b) + iv(a, b) = |z| = \sqrt{a^2 + b^2}$, where $z = a + ib$. We observe that the Cauchy-Riemann equations are not satisfied:

$$\frac{\partial u(a, b)}{\partial a} = \frac{a}{\sqrt{a^2 + b^2}} \neq 0 = \frac{\partial v(a, b)}{\partial y},$$

and, therefore, the derivative does not exist. However, the derivative does not exist even in the real-valued case, due to the non-differentiability of the absolute value function at zero. In our case, we can obtain a necessary condition for a minimal value by zeroing the partial derivatives with respect to the real and imaginary terms.

Recall our main problem (44). In the case where the R norm is an L_2 norm, we can take advantage of the fact that F is a unitary transform (DFT):

$$\hat{\underline{c}} = \arg \min_{\underline{c}} \frac{1}{2} \left\| \frac{1}{\alpha} \underline{g} - \frac{1}{\alpha} (1 - \alpha) F_{\theta}^{-1} \underline{c} \right\|^2 + \lambda \|\underline{c}\|_1. \quad (45)$$

Noting that $\underline{c}_0 = \frac{1}{\alpha} F_{\theta} \underline{g}$, the solution of (45) is obtained by solving K scalar equations of the form:

$$\hat{c}^i = \arg \min_c l := \frac{1}{2} \left| c_0^i - \frac{1}{\alpha} (1 - \alpha) c^i \right|^2 + \lambda |c^i|,$$

where $c_0^i = \xi_0 + j\eta_0$ and $c^i = \xi + j\eta$ are complex numbers. Therefore:

$$\begin{aligned} l &= \frac{1}{2} \left| c_0^i - \frac{1}{\alpha} (1 - \alpha) c^i \right|^2 + \lambda |c^i| \\ &= \frac{1}{2} \left(\xi_0 - \frac{1}{\alpha} (1 - \alpha) \xi \right)^2 + \frac{1}{2} \left(\eta_0 - \frac{1}{\alpha} (1 - \alpha) \eta \right)^2 + \lambda \sqrt{\xi^2 + \eta^2}. \end{aligned} \quad (46)$$

Denote $|c^i| = \sqrt{\xi^2 + \eta^2}$. A necessary condition for extremum is obtained by:

$$\begin{cases} \frac{\partial l}{\partial \xi} = \frac{1}{\alpha} (1 - \alpha) (\xi_0 - \frac{1}{\alpha} (1 - \alpha) \xi) + \frac{\lambda}{|c^i|} \xi = 0 \\ \frac{\partial l}{\partial \eta} = \frac{1}{\alpha} (1 - \alpha) (\eta_0 - \frac{1}{\alpha} (1 - \alpha) \eta) + \frac{\lambda}{|c^i|} \eta = 0. \end{cases} \quad (47)$$

We, therefore, obtain a set of two non-linear, coupled equations for each entry in \underline{c} , and the solution of which is a candidate for the solution to the minimization problem. Fortunately, these equations (47) can be solved analytically, and the following soft-thresholding-based solution is obtained (See Appendix D; (60)):

$$\eta = \begin{cases} \frac{\beta z_0 - \lambda \gamma_0}{\beta^2 \gamma_0^2}, & \frac{\beta z_0}{\gamma_0} > \lambda \\ 0, & \left| \frac{\beta z_0}{\gamma_0} \right| \leq \lambda \\ \frac{\beta z_0 + \lambda \gamma_0}{\beta^2 \gamma_0^2}, & -\lambda > \frac{\beta z_0}{\gamma_0}, \end{cases} \quad (48)$$

and $\xi = \gamma\eta$, where $z_0 = \eta_0 + \gamma\xi_0$, $\gamma_0 = \sqrt{1 + \gamma^2}$ and $\gamma = \frac{\eta_0}{\xi_0}$. λ is a threshold value that needs to be determined.

5.3.4 The non-unitary case

Unfortunately, the norm in our case is not L_2 , but scaled by the fBm covariance inverse matrix (43). Using $\Phi\Phi^T = R^{-1}$ (see also Section 2.2.1 for use in cases of non-invertible matrices) we get:

$$\begin{aligned} \|\underline{g} - F^{-1}\underline{c}\|_R^2 &= (\underline{g} - F^{-1}\underline{c})^T \Phi\Phi^T (\underline{g} - F^{-1}\underline{c}) \\ &= (\Phi^T (\underline{g} - F^{-1}\underline{c}))^T (\Phi^T (\underline{g} - F^{-1}\underline{c})). \end{aligned}$$

Recall that in the unitary case, we use the following derivation:

$$\begin{aligned} \|F^{-1} (F\underline{g} - \underline{c})\| &= (F^{-1} (F\underline{g} - \underline{c}))^H (F^{-1} (F\underline{g} - \underline{c})) \\ &= (F\underline{g} - \underline{c})^H (F^{-1})^H F^{-1} (F\underline{g} - \underline{c}) = \|F\underline{g} - \underline{c}\|. \end{aligned}$$

Therefore, the unitary case requires the following condition: $(F^{-1})^H F^{-1} = I$. In our case, however, this is not satisfied:

$$\begin{aligned} \Phi^T (\underline{g} - F^{-1}\underline{c}) &= \Phi^T F^{-1} (F\underline{g} - \underline{c}) \\ (\Phi^T F^{-1})^H (\Phi^T F^{-1}) &= F\Phi\Phi^T F^{-1}, \end{aligned}$$

rendering the case to be non-unitary.

5.3.5 A Separable Surrogate Functionals (SSF) approach

In the non-unitary case, we turn to one of the available schemes proposed to solve similar minimization problems. Since our problem differs from the usual sparsity-rewarding formulation due to the complex vector, \underline{c} , we reiterate the algorithm stages with the required changes. This follows the formulation in [34], Chapter 6.3.1.

Consider the following cost function:

$$\begin{aligned}
f(\underline{c}) &= \frac{1}{2} \left\| \frac{1}{\alpha} \underline{g} - \frac{1}{\alpha} (1 - \alpha) F^{-1} \underline{c} \right\|_R^2 + \lambda \mathbf{1}^T \rho(\underline{c}) \\
&= \frac{1}{2} \left\| \frac{1}{\alpha} \Phi^T \underline{g} - \frac{1}{\alpha} (1 - \alpha) \Phi^T F^{-1} \underline{c} \right\|^2 + \lambda \mathbf{1}^T \rho(\underline{c}) \\
&\triangleq \frac{1}{2} \|\underline{b} - A\underline{c}\|^2 + \lambda \mathbf{1}^T \rho(\underline{c}),
\end{aligned}$$

where $\underline{b} = \frac{1}{\alpha} \Phi^T \underline{g}$, $A = \frac{1}{\alpha} (1 - \alpha) \Phi^T F^{-1}$, and \underline{c} is a complex vector. Let $d(\underline{c}, \underline{c}_0)$ denote the following distance term:

$$d(\underline{c}, \underline{c}_0) = \frac{\alpha}{2} \|\underline{c} - \underline{c}_0\|^2 - \frac{1}{2} \|A\underline{c} - A\underline{c}_0\|^2,$$

where α_c is chosen so that $\alpha_c I - A^H A \succ 0$, a condition satisfied by choosing $\alpha_c > \lambda_{max}(A^H A)$. The new objective function is:

$$\begin{aligned}
\tilde{f}(\underline{c}) &= \frac{1}{2} \|\underline{b} - A\underline{c}\|^2 + \lambda \mathbf{1}^T \rho + \frac{\alpha_c}{2} \|\underline{c} - \underline{c}_0\|^2 - \frac{1}{2} \|A\underline{c} - A\underline{c}_0\|^2 \\
&= \beta_0 - \underline{c}^H (A^H (\underline{b} - A\underline{c}_0) + \alpha_c \underline{c}_0) + \lambda \mathbf{1}^T \rho(\underline{c}) + \frac{\alpha_c}{2} \|\underline{c}\|^2,
\end{aligned} \tag{49}$$

where β_0 is a constant that does not affect the optimization. Let $\underline{v}_0 = \frac{1}{\alpha_c} A^H (\underline{b} - A\underline{c}_0) + \underline{c}_0$. Substituting \underline{v}_0 in the previous equation (49), we get:

$$\tilde{f}(\underline{c}) = \beta_1 + \frac{\lambda}{\alpha_c} \mathbf{1}^T \rho(\underline{c}) + \frac{1}{2} \|\underline{c} - \underline{v}_0\|^2,$$

where β_1 is a constant. This function presents an optimization problem whose solution is obtained by the same manner as in the unitary case (47) for $\rho(\underline{c}) = |\underline{c}|$. Therefore, optimization of $\tilde{f}(\underline{c})$ is obtained by K scalar optimization problems of the following form:

$$\begin{aligned}
\hat{c}^i &= \arg \min_c l := \frac{1}{2} |c^i - v_0^i|^2 + \frac{\lambda}{\alpha_c} |c^i| \\
&= \arg \min \frac{1}{2} [(\xi^i - \xi_{v_0}^i)^2 + (\eta^i - \eta_{v_0}^i)^2] + \frac{\lambda}{\alpha_c} |c^i|,
\end{aligned}$$

where $v_0^i = \xi_{v_0}^i + j\eta_{v_0}^i$. A necessary condition for obtaining extremum is:

$$\begin{cases} \frac{\partial l}{\partial \xi^i} = (\xi^i - \xi_{v_0}^i) + \frac{\lambda}{\alpha_c |c^i|} \xi^i = 0 \\ \frac{\partial l}{\partial \eta^i} = (\eta^i - \eta_{v_0}^i) + \frac{\lambda}{\alpha_c |c^i|} \eta^i = 0. \end{cases} \tag{50}$$

This adapted version of the SSF algorithm is performed similarly to the original one, in which the shrinkage step is replaced by the solution of a set of non-linearly coupled equations (50). The rest of the steps are performed similarly to the SSF algorithm [34].

Algorithm 2 Image decomposition

1. First iteration ($n = 0$): estimate initial values for $H^{(0)}$ and $\alpha^{(0)}$ (51).
 2. Solve the modified SSF problem (42) and obtain $\underline{h}^{(n)}$, the structured image.
 3. Produce the stochastic image, $\underline{s}^{(n)}$ and calculate $\alpha^{(n+1)}$ by minimizing the L_1 norm of $F\underline{h}^{(n)}$ (52).
 4. Calculate the stochastic image, $\Delta^{(n)} = \underline{g} - (1 - \alpha^{(n+1)}) \underline{h}^{(n)}$, and calculate its Hurst parameter, $H^{(n+1)}$.
 5. Repeat steps 2-4 until the change in estimated parameters is small enough.
-

5.4 Application in texture decomposition

We show a direct application of the aforementioned model. Practically, we observe that in the noiseless case, \underline{g} can be decomposed into its structured and stochastic counterparts, even if the true value of H is unknown (and $H = 0.5$ is assumed). Further, recall the required information for estimation of the mixing parameter, α (35). We propose an iterative image decomposition algorithm (Algorithm 2) that sequentially performs decomposition and parameter estimation.

Since this algorithm is non-convex, the initial guess of the parameters, α and H , should be considered carefully. The initial value of α is obtained as follows: a wavelet-based fractal estimation is performed, yielding the variance-scale plot of the image. Then, $\alpha^{(0)}$ is derived via the R^2 value for the linear fit as follows:

$$\alpha^{(0)} = 4 \cdot \left[\max \left\{ \frac{3}{4} + \frac{\epsilon}{4}, R^2 \right\} - \frac{3}{4} \right], \quad (51)$$

where ϵ is a small number. Since R^2 indicates the goodness of fit, it shows the overall fractal behavior of the given image, and therefore, it should correspond to the mixing parameter, α . For instance, high R^2 values indicate a values of α close to 1. We consider $R^2 < \frac{3}{4}$ to indicate non-fractal behaviour, for which the initial guess for α shall be ϵ . $H^{(0)}$, the initial value of H , is given by the fractal dimension of the given image.

Due to the non-convexity of the algorithm, we found empirically that a correct estimation of α via the optimal estimator (35) gives unsatisfactory results. Instead, for each iteration, we produce $\hat{\underline{h}}$, the estimated structural image:

$$\hat{\underline{h}} = \frac{1}{1 - \hat{\alpha}} [\underline{g} - \hat{\alpha} \hat{\underline{s}}],$$

where $\hat{\underline{s}}$ is the decomposed stochastic image for the current iteration. Then, we iterate through values of $\hat{\alpha} \in [0, 1]$, and choose the value that maximizes the L_1 norm of the transform's coefficients of $\hat{\underline{h}}$:

$$\hat{\alpha} = \arg \min_{\alpha \in [0, 1]} \left\| F \hat{\underline{h}} \right\|_1. \quad (52)$$

5.4.1 Global, local and non-local texture decomposition algorithms

The texture model is valid if the strongest local frequencies can be composed into a single set of globally-strongest frequencies. This is the basic assumption of the texture model and the decomposition algorithm. However, in many textures, the leading frequencies may change throughout the image. Therefore, two variants for the decomposition algorithm are proposed, a local and a non-local algorithm, to complement the global decomposition algorithm.

The local algorithm assumes globally strongest frequencies to be valid only in a local neighborhood of a certain patch. The set of globally-strongest patches in this case is, therefore, not a single set across the image, but changes with the spatial location. A typical neighborhood size shall be a margin of 50% from each direction of a certain patch. This algorithm captures locally varying information better, but does not exploit global structures if they are present in a textured image.

The non-local algorithm combines the local and the global approaches. First, the texture is divided into several non-overlapping regions that span the entire image, where each region contains patches of similar textural and fractal characteristics. Then, for each region, a globally-strongest set of frequencies is estimated, similarly to the global case. These regions are created by extracting structural and fractal features from a specific patch: the structural features are the contrast, correlation, energy and homogeneity Haralick features from the grey-level co-occurrence matrix [35], and the fractal features are the H value, the R^2 of the log-linear fit for the variance/scale plot, and the polynomial fitting parameters for a third-order polynomial fitting of the variance/scale plot of the patch. These features then undergo feature reduction by means of PCA for 3 dimensions, and the regions are the clustering labels obtained by K-means, applied on the resulting PCA coefficients.

The non-local texture decomposition algorithm has the advantage of extracting local features when a certain area of the image has different textural characteristics than other areas, while still exploiting large segments of the textured image to extract the globally-strongest frequencies. It is, therefore, the algorithm used in the following simulations and examples.

5.4.2 Simulated examples

To evaluate the algorithm, we first present a series of experiments based on manual composition of the stochastic and structured images. The simulated values spanned the reasonable range $H \in [0.2, 0.8]$ and $\alpha \in [0.2, 0.8]$, from which estimation of these parameters, as well as the decomposition of the images, were obtained. We repeated the experiment 5 times for each set of (H, α) , in which the stochastic image was simulated randomly. The total number of experiments was 245. Each obtained image then underwent decomposition (Alg. 2) to yield $\hat{\alpha}$, \hat{H} , and the two reconstructed images.

The experiment was assessed via the average MSE values for $\hat{\alpha}$ and \hat{H} , and the SSIM for the image reconstruction, comparing to the ground truth. We note (Fig. 20) the following remarks regarding the simulations' results: the stochastic image was reconstructed better for higher values

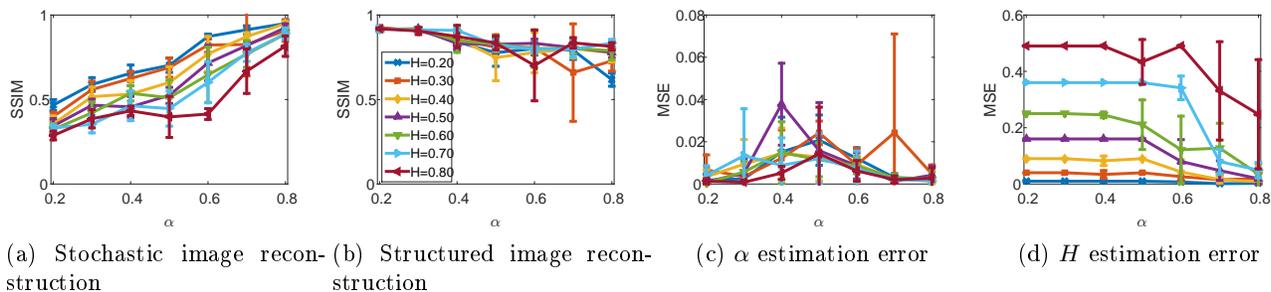


Figure 20: Decomposition simulation results: (20a), (20b) SSIM reconstruction results of the stochastic and structured images, respectively. (20c), (20d) MSE of the estimated α and H values, respectively. All figures depict the simulation result for experiments with various values of α and H (indicated by the legend (20b)), where each point represents an average of 5 experiments (with corresponding error bars depicting the standard deviation). We observe that the mixing and fractal parameters (α and H , respectively) have been estimated with low MSE. The stochastic image reconstruction quality is better for higher values of α , indicating higher stochastic content, and the structural image reconstruction depicts the expected, inverse, outcome.

of α , and vice versa - the structured image was reconstructed better for lower values of α ; the overall estimation quality of α shows low MSE, indicating satisfactory estimation; the estimation of H shows reasonable results in high values of α , or in low values of H . This is due to the fact that high values of H depict an image with smooth behaviour (with LRD), which is harder to estimate in images with small spatial size.

To further demonstrate the quality of the algorithm, we analyze one of the examples used in the simulation (Fig. 21), with the parameter set ($\alpha = 0.4, H = 0.3$). We observe that despite the relatively low SSIM value for the stochastic image (0.634), the two images appear to have been reconstructed correctly, despite the challenging initial simulated image (Fig. 21a). Gaussianity (Fig. 21e) and fractal behavior (Fig. 21h) are observed in the stochastic images, whereas sparsity is observed in the transform coefficients of the structural image (Fig. 21i). We note that due to artifacts near edges, the reconstructed stochastic image underwent edge-tapering by a 3×3 neighborhood low-pass filter.

5.4.3 Examples with natural images

We evaluate the proposed algorithm using natural images (Fig. 22 shows an elaborate example of the fractal extraction) and compare the proposed decomposition algorithm to two state-of-the-art texture-cartoon decomposition methods, TV-G [36, 37] and MCA [38, 39]. This is not a completely compatible comparison, since the purpose of the proposed decomposition method is different; our method separates a texture into a stochastic and structural part, whereas the other methods decompose a general image into its cartoon-type skeleton and its texture. Therefore, in TV-G, we

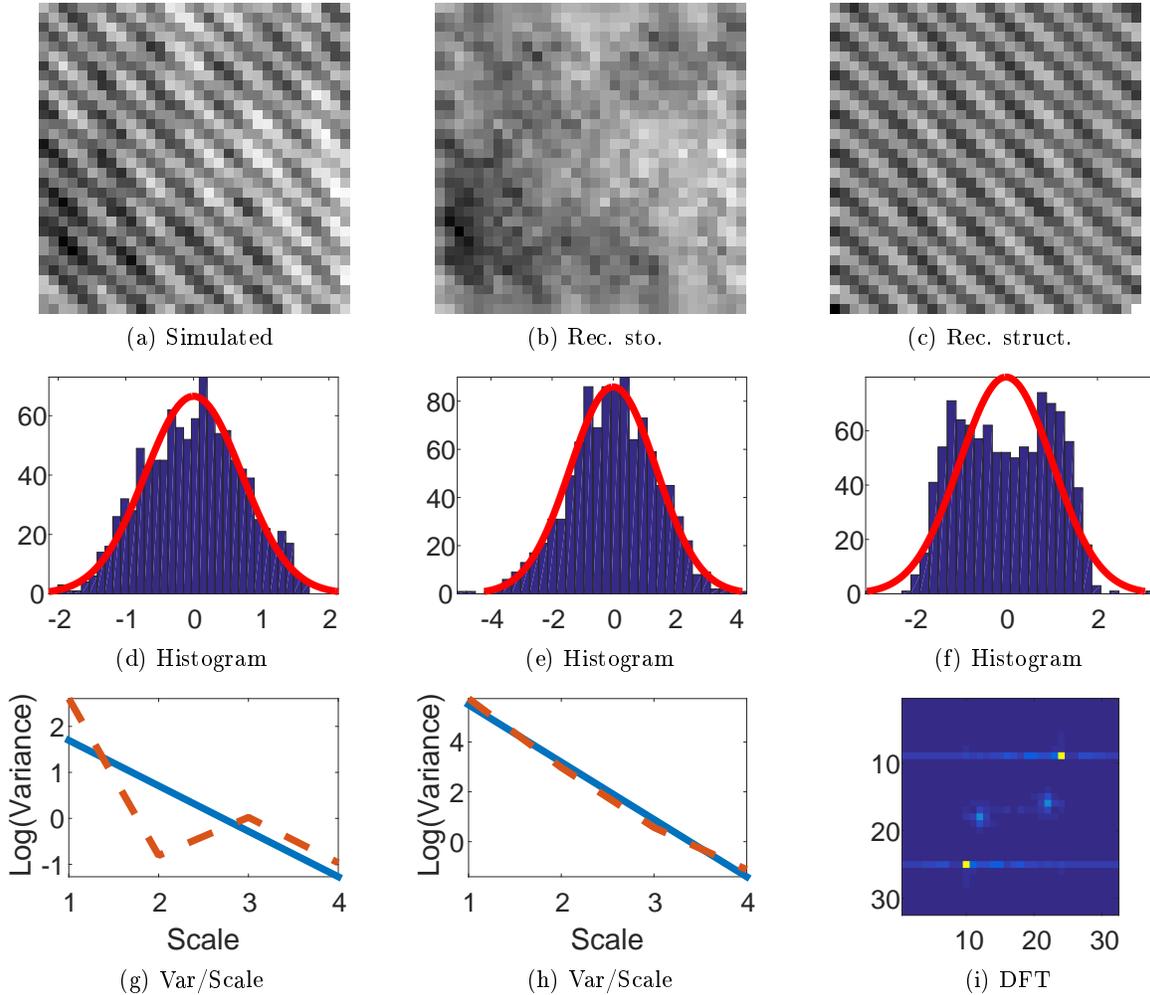


Figure 21: Decomposition example: (21a), (21b) and (21c): ground truth, reconstructed stochastic (SSIM 0.634) and structured (SSIM 0.797) images, respectively, with their corresponding histograms, (21d), (21e) and (21f), respectively. The stochastic image shows Gaussian and fractal behavior (21h), relative to the ground truth (21g), whereas the structured image is sparse in the transform domain (21i).

manually choose appropriate parameters (for cartoon and texture regularization) that yield the resulting two layers. In MCA, we consider the stochastic part to be the image residual, and the structured part to be the sum of the cartoon and texture. Whereas there exists a paper with a similar application [40], a code for providing comparisons could not be obtained.

A comparison was performed on several textures from the KTH-TIPS2b dataset (Fig. 23, first row). Observe that the structural parts of the proposed algorithm (Fig. 23, second row) is much more visually clear and free of noise relative to the results obtained by using the other algorithms (Fig. 23, third and fourth rows). While the images do not contain noisy or stochastic details, the

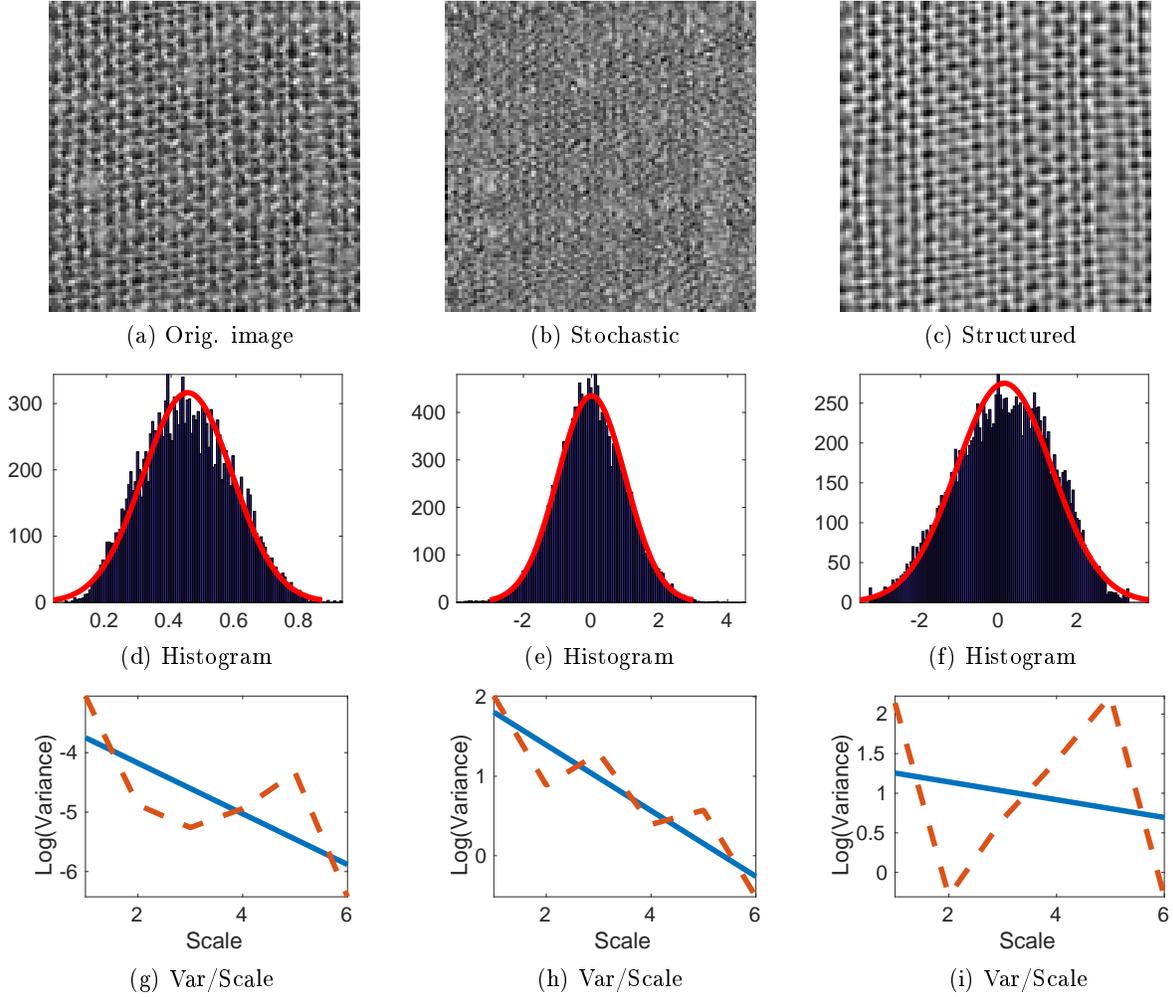


Figure 22: Example of image decomposition: 22a original image, 22b stochastic image and 22c structured image with their corresponding histograms, 22d, 22e and 22f, and variance/scale plots, 22g, 22h and 22i, respectively. The R^2 for the variance/scale fit of the original image, stochastic image and structured image were 0.52, 0.82 and 0.03, and the KL-divergences (w.r.t normal distribution) were 0.0172, 0.0036 and 0.0253, respectively. Observe the enhanced sparsity of the structured image and enhanced Gaussianity and fractal behavior of the stochastic image, relative to the original image.

quality of the structured edges in the image has been retained, and has not been over-emphasized.

Comparing the stochastic parts of the same images (Fig. 24), we note that the result of the proposed algorithm (Fig. 24, first row) yields surfaces close to fractal, depicting some long-range dependencies (it is especially apparent in texture in the second column), distinctive of fractal surfaces. We observe that comparing with the other algorithms (Fig. 24, second and third rows), the stochastic layer contains significant information (and does not resemble white noise), while not

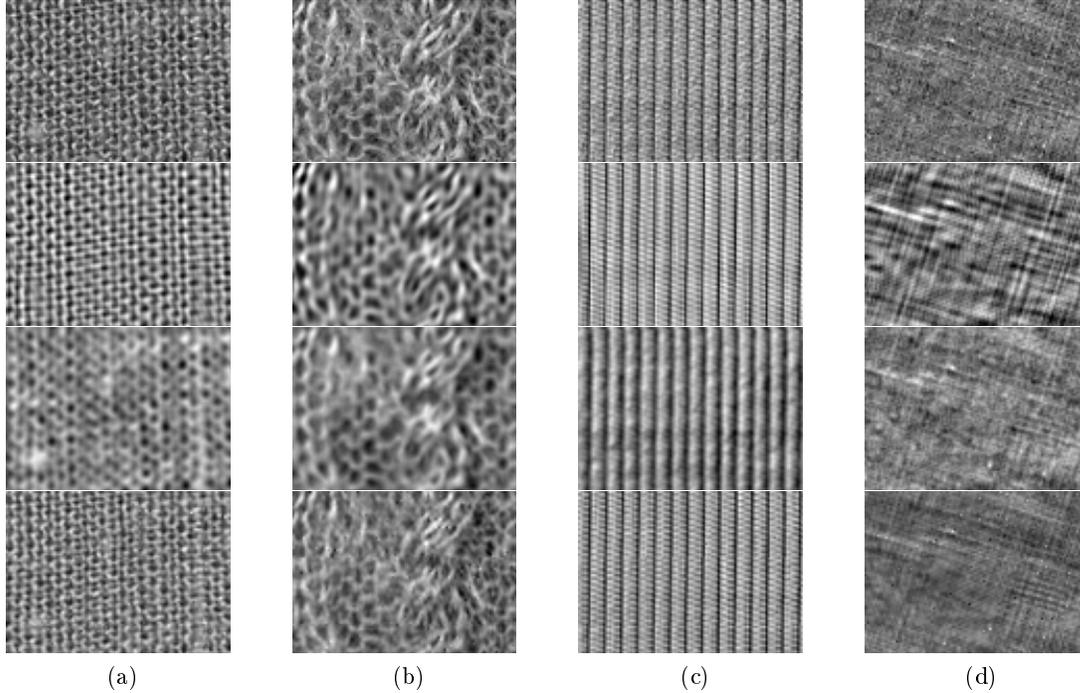


Figure 23: Decomposition comparison. Columns depict different textures from the KTH-TIPS2b dataset. The first row depicts the ground truth; rows 2–4 depict the structural part using the proposed algorithm, TV-G and MCA, respectively.

containing structural information. It may, therefore, be concluded that it has successfully extracted the fractal details from the textures. This is also corroborated by the fractal fit R^2 values, shown in the caption of Fig. 24.

The proposed decomposition algorithm does not separate images into a high- and low-pass-type images. The separated fractal image part clearly contains both high and low frequencies, and the emphasis of the decomposition is on the suitable representation of each image component. We note that while the sparsity-based method provides only an approximated solution, an exact solution, optimal in the MMSE-sense, is not tractable using a direct approach (Appendix E).

6 Discussion

The phase of stochastic textures is an elusive subject. While it has a prominent structure, pronounced non-locally in the harmonic decomposition of the periodic ingredients of textures, as well as in anisotropic, orientational sharp details, it is useful in many tasks to retain the phase and process only the magnitude, due to robustness of phase to blur and noise degradations. Further, the concept of phase, while usually discussed in the Fourier domain, can be advantageous in its incorporation into recently-proposed more recent transforms, such as the complex steerable wavelets

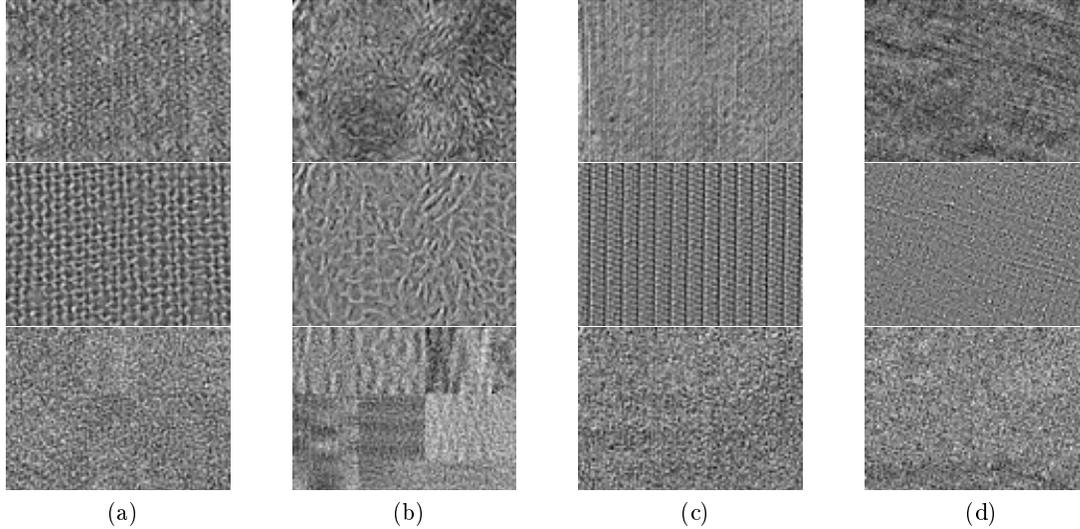


Figure 24: Decomposition comparison of the stochastic part of the textures in Fig. 23. Columns correspond to the textures in Fig. 23. Rows 1–3 depict the complementary stochastic part, using the proposed algorithm, TV-G and MCA, respectively. Fractal fit R^2 : Row 1 (proposed), left to right: 0.82, 0.92, 0.85 and 0.96; row 2 (TV-G): 0.02, 0.37, 0.57 and 0.98; row 3 (MCA): 0.30, 0.60, 0.59 and 0.47.

[41].

The utilization of phase in textures has been considered in this work by addressing the magnitude and phase separately for image deconvolution and decomposition. We observe that while retaining the phase for deconvolution, successful restoration of details was obtained even for isotropic textures that may not follow the phase model exactly, thus showing the generality of this type of algorithm.

The proposed combined model was used in this study for decomposition of textures. The main optimization decomposition problem (42) differs from the typical sparse coding or basis pursuit denoising-type problems, in which an image is decomposed into a sparsely represented image (or several images, in the case of morphological component analysis) and a noise image. The noise is usually discarded as non-informative interference. In our case, however, the objective is separation of a given image into two conceptually different but meaningful ingredients: a sparsely-represented image and a densely-represented image, in our case - an fBm/fGn image. Our formulation is general and can be applied to any linear transform applied on a random vector.

The proposed model can also be used for further image processing tasks such as super-resolution or deconvolution. This is due to the fact that the model separates the image content conceptually into two relatively straightforward structures: a well-defined random process and a harmonic sequence. In the case of super-resolution, for instance, each of these ingredients can be interpolated with known methods.

References

- [1] Jean François Aujol, Gilles Aubert, Laure Blanc-Féraud, and Antonin Chambolle. Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vis.*, 22(1):71–88, 2005.
- [2] Thrasyvoulos N. Pappas, David L. Neuhoff, Huib de Ridder, and Jana Zujovic. Image Analysis: Focus on Texture Similarity. *Proc. IEEE*, 101(9):2044–2057, sep 2013.
- [3] N R Mudigonda, R M Rangayyan, and J E Desautels. Detection of breast masses in mammograms by density slicing and texture flow-field analysis. *IEEE Trans. Med. Imaging*, 20(12):1215–27, dec 2001.
- [4] Wen-Chieh Lin, James Hays, Chenyu Wu, Vivek Kwatra, and Yanxi Liu. A comparison study of four texture synthesis algorithms on regular and near-regular textures. Technical report, Carnegie Mellon University, 2004.
- [5] I Zachevsky and Y Y Zeevi. Single-Image Superresolution of Natural Stochastic Textures Based on Fractional Brownian Motion. *IEEE Trans. Image Process.*, 23(5):2096–2108, 2014.
- [6] A Srivastava, A B Lee, and E P Simoncelli. On Advances in Statistical Modeling of Natural Images. *J. Math. Imaging Vis.*, 18:17–33, 2003.
- [7] Ido Zachevsky and Yehoshua Y Zeevi. On the statistics of natural stochastic textures and their application in image processing. In *IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 5829–5833, Florence, Italy, may 2014.
- [8] Ido Zachevsky and Yehoshua Y Zeevi. Superresolution of self-similar textures. *CCIT Report. EE Pub, Tech. Isr. Inst. Technol.*, 838(1795), 2013.
- [9] Ido Zachevsky and Yehoshua Y Zeevi. Statistics of Natural Stochastic Textures and Their Application in Image Denoising. *IEEE Trans. Image Process.*, 25(May):2130–2145, 2016.
- [10] Ido Zachevsky and Yehoshua Y. Zeevi. Model-based Color Natural Stochastic Textures Processing and Classification. In *IEEE Glob.*, pages 1357–1361, Orlando, FL, dec 2015.
- [11] Ido Zachevsky and Yehoshua Y Zeevi. Combining long-range dependencies with phase information in Natural Stochastic Texture enhancement. In *IEEE Int. Conf. Image Process.*, Paris, France, oct 2014.
- [12] Alan V. Oppenheim and Jae S. Lim. Importance of Phase in Signals. *Proc. IEEE*, 69(5):529–541, 1981.
- [13] Joan Jacobs, Nikolay Skarbnik, Y Yehoshua, Chen Sagiv, and Yehoshua Y Zeevi. The Importance of Phase in Image Processing The Importance of Phase in Image Processing. (August), 2010.
- [14] Bruno Galerne, Yann Gousseau, and Jean Michel Morel. Random phase textures: Theory and synthesis. *IEEE Trans. Image Process.*, 20(1):257–267, 2011.
- [15] Bela Julesz. Visual pattern discrimination. *Inf. Theory, IRE Trans.*, 49:41–46, 1962.
- [16] Javier Portilla and Eero P Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. 40(1):49–71, 2000.
- [17] A. P. Pentland. Fractal-based description of natural scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):661–74, jun 1984.

- [18] Benoit B Mandelbrot and John W Van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM Rev.*, 10(4):422–437, 1968.
- [19] Gennady Samorodnitsky. Long memory and self-similar processes. *Ann. - Fac. des Sci. Toulouse Math.*, 15(1):107–129, 2006.
- [20] Beatrice Pesquet-Popescu and Jacques L Vehel. Stochastic fractal models for image processing. *IEEE Signal Process. Mag.*, 19(5):48–62, sep 2002.
- [21] Romain Francois Peltier and Jacques Levy-vehel. Multifractional Brownian motion: definition and preliminary results. Technical report, INRIA, 1995.
- [22] Pierrick Legrand and Jacques Levy Vehel. Signal and Image processing with FracLab. *Fractal*, pages 4–7, 2004.
- [23] Emmanuel Perrin, Rachid Harba, Ileana Iribarren, and Rachid Jennane. Piecewise fractional Brownian motion. 53(3):1211–1215, 2005.
- [24] a. Gupta and S. Joshi. Some Studies on the Structure of Covariance Matrix of Discrete-Time fBm. *IEEE Trans. Signal Process.*, 56(10):4635–4650, oct 2008.
- [25] Li Li, Jianming Hu, Yudong Chen, and Yi Zhang. PCA Based Hurst Exponent Estimator for fBm Signals Under Disturbances. *IEEE Trans. Signal Process.*, 57(7):2840–2846, jul 2009.
- [26] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.
- [27] John F. Kenney. *Mathematics of Statistics, Part II*, 1940.
- [28] Michael Elad and Mario A T Figueiredo. On the Role of Sparse and Redundant Representations in Image Processing. *Proc. IEEE*, 98(6):972–982, jun 2010.
- [29] Martin Welk, David Theis, Thomas Brox, and Joachim Weickert. PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. *Scale Sp. PDE Methods Comput. Vis.*, pages 585–597, 2005.
- [30] Zhou Wang and Alan C Bovik. Mean squared error: Love It or Leave It? *IEEE Signal Process. Mag.*, pages 98–117, jan 2009.
- [31] Phil Brodatz. *Textures: a photographic album for artists and designers*, volume 66. Dover New York, 1966.
- [32] Ido Zachevsky and Yehoshua Y. Zeevi. On the statistics of Natural Stochastic Textures. *CCIT Report. EE Pub, Tech. Isr. Inst. Technol.*, 862(1819), 2014.
- [33] Ido Zachevsky and Yehoshua Y. Zeevi. On the phase of natural stochastic textures with application in deconvolution and image decomposition. Technical report, Technion - IIT, Haifa, Israel, 2016.
- [34] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Number 1. 2010.
- [35] Robert M Haralick. Statistical and Structural Approaches to Texture. 67(5), 1979.

- [36] Antonin Chambolle and Jean-françois Aujol. Dual norms and image decomposition models. (March), 2004.
- [37] J Gilles and Stanley Osher. Bregman Implementation of Meyer’s G-Norm for Cartoon+ Textures Decomposition. 1:1–5, 2011.
- [38] M. Elad, J.-L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Appl. Comput. Harmon. Anal.*, 19(3):340–358, nov 2005.
- [39] M Fadili, Jean-luc Starck, J Bobin, and Yassir Moudden. Image Decomposition and Separation Using Sparse Representations: An Overview. *Proc. IEEE*, 98(6):983–994, 2010.
- [40] Yong Huang and Kap Luk Chan. Texture Decomposition by Harmonic Extraction From Higher Order Statistics. 13(1):1–14, 2004.
- [41] Neal Wadhwa and Michael Rubinstein. Phase-based video motion processing. *ACM Trans. ...*, 32(4):80:1–80:10, 2013.

Appendices

A The variance of the sample variance of correlated Gaussian variables

Let X be a correlated multivariate Gaussian vector. There exist a matrix A and a vector m such that

$$X = AZ + m,$$

where Z is an i.i.d Gaussian vector with zero mean and unit variance. The sample variance of X is given by:

$$s^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2,$$

and the variance of the sample variance is then given by:

$$\begin{aligned} \mathbb{E} \left\{ (s^2 - \mathbb{E} \{s^2\})^2 \right\} &= \mathbb{E} \left\{ (s^2)^2 \right\} - \mathbb{E} \{s^2\}^2 \\ &= \frac{1}{N^2} \mathbb{E} \left\{ \left((X - \bar{X})^T (X - \bar{X}) \right)^2 \right\} - \frac{1}{N^2} \mathbb{E} \left\{ (X - \bar{X})^T (X - \bar{X}) \right\}^2 \\ \mathbb{E} \left\{ \left((X - \bar{X})^T (X - \bar{X}) \right)^2 \right\} &= \mathbb{E} \left\{ \left((AZ)^T (AZ) \right)^2 \right\} = \mathbb{E} \left\{ (Z^T A^T A Z)^2 \right\} \\ &= \mathbb{E} \left\{ (Z^T \Lambda Z)^2 \right\}. \end{aligned}$$

For the calculation of $Z^T \Lambda Z$, assume $Z = (z_1, z_2, \dots, z_N)^T$ and $\Lambda = (\lambda_{i,j})$. Then:

$$\begin{aligned}
Z^T \Lambda Z &= \begin{pmatrix} z_1 \\ \vdots \\ z_N \end{pmatrix} \begin{pmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1N} \\ \vdots & \ddots & & \\ \lambda_{N1} & & & \lambda_{NN} \end{pmatrix} (z_1, \dots, z_N) \\
&= \begin{pmatrix} z^T \lambda_{\cdot 1} \\ \vdots \\ z^T \lambda_{\cdot N} \end{pmatrix}^T (z_1, \dots, z_N) \\
&= \sum_i z_i \cdot z^T \lambda_{\cdot i} \\
&= \sum_{i,j} z_i \cdot z_j \lambda_{ji},
\end{aligned}$$

where $\lambda_{\cdot i}$ denotes the i th column of Λ . Therefore,

$$\begin{aligned}
\mathbb{E} \{ Z^T \Lambda Z \} &= \mathbb{E} \left\{ \sum_{ij} z_i z_j \lambda_{ji} \right\} = \sum_{ij} \lambda_{ji} \mathbb{E} \{ z_i z_j \} \\
&= \sum_{ij} \lambda_{ji} \delta(i, j) = \sum_i \lambda_{ii} = \text{Tr} \{ \Lambda \},
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E} \left\{ (Z^T \Lambda Z)^2 \right\} &= \mathbb{E} \{ Z^T \Lambda Z \cdot Z^T \Lambda Z \} \\
&= \mathbb{E} \left\{ \sum_{i,j} z_i z_j \lambda_{ji} \cdot \sum_{k,l} z_k z_l \lambda_{lk} \right\} \\
&= \mathbb{E} \left\{ \sum_{k,l} \sum_{i,j} z_i z_j z_k z_l \lambda_{ji} \lambda_{lk} \right\}.
\end{aligned}$$

Recall that

$$\begin{aligned}
\mathbb{E} \{ z_i z_j z_k z_l \} &= \mathbb{E} \{ z_i z_j \} \mathbb{E} \{ z_k z_l \} + \mathbb{E} \{ z_i z_l \} \mathbb{E} \{ z_k z_j \} + \mathbb{E} \{ z_i z_k \} \mathbb{E} \{ z_j z_l \} \\
&= \delta(i, j) \delta(k, l) + \delta(i, l) \delta(k, j) + \delta(i, k) \delta(j, l),
\end{aligned}$$

therefore:

$$\begin{aligned}
\mathbb{E} \left\{ (Z^T \Lambda Z)^2 \right\} &= \sum_{k,l} \sum_{i,j} \lambda_{ji} \lambda_{lk} \mathbb{E} \{ z_i z_j z_k z_l \} \\
&= \sum_{i,k} \lambda_{ii} \lambda_{kk} + 2 \sum_{i,j} \lambda_{ij}^2 \\
&= \sum_{i,j} \lambda_{ii} \lambda_{jj} + 2 \sum_{i,j} \lambda_{ij}^2 \\
&= \text{Tr} \{ \Lambda \}^2 + 2 \text{Tr} \{ \Lambda^2 \}.
\end{aligned}$$

Using the aforementioned calculation, the variance of the sample variance is given by:

$$\text{Var} \{ s^2 \} = \frac{2}{N^2} \text{Tr} \{ \Lambda^2 \}, \quad (53)$$

and is, therefore, approaching zero in quadratic rate.

B The shifted-phase DCT matrix

The shifted-phase DCT matrix, D_θ , is defined via the 1D-DCT matrix, D_θ^1 , for $0 \leq k \leq N-1$, $0 \leq n \leq N-1$ as follows:

$$[D_\theta^1]_{k,n} = \sqrt{\frac{2}{N}} w_k \cos \left[\frac{\pi}{2N} (2n+1)k + 2\pi\theta_n \right],$$

where

$$w_k = \begin{cases} \frac{1}{\sqrt{1+\cos(4\pi\theta_0)}}, & k=0 \\ 1, & k>0, \end{cases} \quad (54)$$

and $0 \leq \theta_n \leq 1$ depends on the frequency, n . According to the phase model, a different phase may be added (according to the absolute patch position in the image, or at least with respect to adjacent patches) to each frequency.

The standard DCT does not encode phase information, and in that sense it is not invariant to phase shifts, as a shifted signal would have a different decomposition of zero-phase waves (c.f. DFT, which encodes phase information). The shifted-phase transform, D_θ , remains an orthogonal matrix, since a phase shift of one harmonic w.r.t the other does not change this property. Specifically, we calculate the inner product between two such basis functions with different frequencies, k and l ,

and different phases, θ_k and θ_l , respectively:

$$(e_k, e_l) = \left(\cos \left(\frac{\pi}{2N} (2n+1)k + 2\pi\theta_k \right), \cos \left(\frac{\pi}{2N} (2n+1)l + 2\pi\theta_l \right) \right) \quad (55)$$

$$= \sum_{n=0}^{N-1} \cos \left(\frac{\pi}{2N} (2n+1)k + 2\pi\theta_k \right) \cos \left(\frac{\pi}{2N} (2n+1)l + 2\pi\theta_l \right) \quad (56)$$

$$= \frac{1}{2} \sum_{n=0}^{N-1} \left[\cos \left(\frac{\pi}{2N} (2n+1)(k+l) + 2\pi(\theta_k + \theta_l) \right) + \cos \left(\frac{\pi}{2N} (k-l) + 2\pi(\theta_k - \theta_l) \right) \right]. \quad (57)$$

Denote $\theta' \triangleq 2\pi(\theta_k + \theta_l)$ and $\tau \triangleq \pi(k+l)$. The first term in (57) is as follows:

$$\begin{aligned} \sum_{n=0}^{N-1} \left[\cos \left(\tau \frac{n}{N} + \frac{\tau}{2N} + \theta' \right) \right] &= \Re \sum_{n=0}^{N-1} \exp \left\{ j \left(\frac{\tau n}{N} + \frac{\tau}{2N} + \theta' \right) \right\} \\ (*) &= \Re e^{j \left(\frac{\tau}{2N} + \theta' \right)} \sum_{n=0}^{N-1} \left(e^{j\tau/N} \right)^n, \end{aligned}$$

and this sum vanishes for $k \neq l$, similarly to the standard DCT. If $k = l$, we get a similar calculation to that of the standard DCT, with $\tau = 2\pi k$ and $\theta' = 4\pi\theta_k$:

$$(*) = \Re e^{j \left(\frac{\tau}{2N} + \theta' \right)} \frac{1 - e^{j2\pi k}}{1 - e^{j2\pi k/N}} = 0.$$

Similarly, the calculation for the second term in (57), with $\theta' = 2\pi(\theta_k - \theta_l)$ and $\tau \triangleq \pi(k-l)$, yields zero for $k \neq l$, and N when $k = l$. In the case of $k = l = 0$, we obtain $\cos(4\pi\theta_k)$ as a multiplicative constant for the first term in (57). Therefore, using w_k as defined in (54), we get orthogonality. We note that the shiftet-phase DCT degenerates to standard DCT for $\theta_k = 0 \forall k$.

C Orthogonality of the non-separable DCT transform

Let (n, m) denote spatial coordinates and (k_x, k_y) denote spatial frequencies. The inner product of two elements, e_l and e_k , is as follows:

$$\begin{aligned}
(e_k, e_l) &= \sum_{n,m} \cos\left(\pi \frac{(2n+1)k_x + (2m+1)k_y}{2N}\right) \cos\left(\pi \frac{(2n+1)l_x + (2m+1)l_y}{2N}\right) \\
&= \sum_{n,m} \left[\cos\left(\pi \frac{(2n+1)k_x}{2N}\right) \cos\left(\pi \frac{(2m+1)k_y}{2N}\right) - \sin\left(\pi \frac{(2n+1)k_x}{2N}\right) \sin\left(\pi \frac{(2m+1)k_y}{2N}\right) \right] \cdot \\
&\quad \cdot \cos\left(\pi \frac{(2n+1)l_x + (2m+1)l_y}{2N}\right) \\
&= \sum_{n,m} \left[\cos\left(\pi \frac{(2n+1)k_x}{2N}\right) \cos\left(\pi \frac{(2m+1)k_y}{2N}\right) - \sin\left(\pi \frac{(2n+1)k_x}{2N}\right) \sin\left(\pi \frac{(2m+1)k_y}{2N}\right) \right] \cdot \\
&\quad \cdot \left[\cos\left(\pi \frac{(2n+1)l_x}{2N}\right) \cos\left(\pi \frac{(2m+1)l_y}{2N}\right) - \sin\left(\pi \frac{(2n+1)l_x}{2N}\right) \sin\left(\pi \frac{(2m+1)l_y}{2N}\right) \right] \\
&= \sum_{n,m} \cos\left(\pi \frac{(2n+1)k_x}{2N}\right) \cos\left(\pi \frac{(2m+1)k_y}{2N}\right) \cos\left(\pi \frac{(2n+1)l_x}{2N}\right) \cos\left(\pi \frac{(2m+1)l_y}{2N}\right) + \\
&\quad - \sum_{n,m} \cos\left(\pi \frac{(2n+1)k_x}{2N}\right) \cos\left(\pi \frac{(2m+1)k_y}{2N}\right) \cdot \sin\left(\pi \frac{(2n+1)l_x}{2N}\right) \sin\left(\pi \frac{(2m+1)l_y}{2N}\right) \\
&\quad - \sum_{n,m} \sin\left(\pi \frac{(2n+1)k_x}{2N}\right) \sin\left(\pi \frac{(2m+1)k_y}{2N}\right) \cos\left(\pi \frac{(2n+1)l_x}{2N}\right) \cos\left(\pi \frac{(2m+1)l_y}{2N}\right) + \\
&\quad + \sum_{n,m} \sin\left(\pi \frac{(2n+1)k_x}{2N}\right) \sin\left(\pi \frac{(2m+1)k_y}{2N}\right) \sin\left(\pi \frac{(2n+1)l_x}{2N}\right) \sin\left(\pi \frac{(2m+1)l_y}{2N}\right) \\
&= \sum_n \cos\left(\pi \frac{(2n+1)l_x}{2N}\right) \cos\left(\pi \frac{(2n+1)k_x}{2N}\right) \sum_m \cos\left(\pi \frac{(2m+1)k_y}{2N}\right) \cos\left(\pi \frac{(2m+1)l_y}{2N}\right) + \\
&\quad - \sum_n \cos\left(\pi \frac{(2n+1)k_x}{2N}\right) \sin\left(\pi \frac{(2n+1)l_x}{2N}\right) \sum_m \cos\left(\pi \frac{(2m+1)k_y}{2N}\right) \sin\left(\pi \frac{(2m+1)l_y}{2N}\right) \\
&\quad - \sum_n \sin\left(\pi \frac{(2n+1)k_x}{2N}\right) \cos\left(\pi \frac{(2n+1)l_x}{2N}\right) \sum_m \sin\left(\pi \frac{(2m+1)k_y}{2N}\right) \cos\left(\pi \frac{(2m+1)l_y}{2N}\right) + \\
&\quad + \sum_n \sin\left(\pi \frac{(2n+1)k_x}{2N}\right) \sin\left(\pi \frac{(2n+1)l_x}{2N}\right) \sum_m \sin\left(\pi \frac{(2m+1)k_y}{2N}\right) \sin\left(\pi \frac{(2m+1)l_y}{2N}\right) \\
&\triangleq S_{cc,x}S_{cc,y} - S_{cs,x}S_{cs,y} - S_{sc,x}S_{sc,y} + S_{ss,x}S_{ss,y} \tag{58}
\end{aligned}$$

From known properties of 1D-DCT he have the following:

$$S_{cc,x} = S_{ss,x} = \begin{cases} 0, & l_x \neq k_x \\ N, & l_x = k_x \end{cases}, S_{cc,y} = S_{ss,y} = \begin{cases} 0, & l_y \neq k_y \\ N, & l_y = k_y. \end{cases}$$

The last equality stems from the orthogonality of the 1D-DCT transform; the inner product of two basis elements of the DCT, (e_k^1, e_l^1) , is zero:

$$\begin{aligned} (e_k, e_l) &= \sum_n \cos\left(\frac{\pi}{2N}(2n+1)k\right) \cos\left(\frac{\pi}{2N}(2n+1)l\right) \\ &= \frac{1}{2} \sum_n \cos\left(\frac{\pi}{2N}(2n+1)(k+l)\right) + \cos\left(\frac{\pi}{2N}(2n+1)(k-l)\right) = 0, \end{aligned}$$

where the last equality follows from the following result, for any $k \neq 0$:

$$\begin{aligned} \sum_n \cos\left(\frac{\pi}{2N}(2n+1)k\right) &= \sum_n \Re e^{j\frac{\pi}{2N}(2n+1)k} = \Re e^{j\frac{\pi}{2N}k} \sum_n e^{j(\frac{\pi}{N}k)n} \\ &= \Re e^{j\frac{\pi}{2N}k} \frac{\sin\left(\frac{1}{2}N\left(\frac{\pi}{N}k\right)\right)}{\sin\left(\frac{1}{2}\left(\frac{\pi}{N}k\right)\right)} e^{j\left(\frac{\pi}{N}k\right)\frac{N-1}{2}} \\ &= \frac{\cos\left(\frac{1}{2}\pi k\right) \sin\left(\frac{1}{2}\pi k\right)}{\sin\left(\frac{1}{2}\frac{\pi k}{N}\right)} = \frac{\frac{1}{2}(\sin 0 + \sin \pi k)}{\sin\left(\frac{1}{2}\frac{\pi k}{N}\right)} = 0, \end{aligned}$$

where $\Re x$ denotes the real part of a complex variable, x . It remains to check the cross terms (58) for orthogonality:

$$\begin{aligned} S_{sc,x} &= \sum_n \sin\left(\frac{\pi}{2N}[(2n+1)k_x]\right) \cos\left(\frac{\pi}{2N}[(2n+1)l_x]\right) \\ &= \frac{1}{2} \sum_n \sin\left(\frac{\pi}{2N}[(2n+1)(k_x+l_x)]\right) + \sin\left(\frac{\pi}{2N}[(2n+1)(k_x-l_x)]\right), \end{aligned}$$

where

$$\begin{aligned} \sum_n \sin\left(\frac{\pi}{2N}(2n+1)k\right) &= \sum_n \Im e^{j\frac{\pi}{2N}(2n+1)k} = \sum_n \Im e^{j\frac{\pi k}{2N}} e^{j\pi\frac{n}{N}k} \\ &= \Im e^{j\frac{\pi k}{2N}} \sum_n e^{j\left(\frac{\pi k}{N}\right)n} = \Im e^{j\frac{\pi k}{2N}} \frac{\sin\left(\frac{1}{2}N\frac{\pi k}{N}\right)}{\sin\left(\frac{1}{2}\frac{\pi k}{N}\right)} e^{j\left(\frac{\pi k}{N}\right)\frac{N-1}{2}} \\ &= \Im \frac{\sin\left(\frac{1}{2}\pi k\right)}{\sin\left(\frac{1}{2}\frac{\pi k}{N}\right)} e^{j\frac{1}{2}\pi k} = \frac{\sin\left(\frac{1}{2}\pi k\right)^2}{\sin\left(\frac{1}{2}\frac{\pi k}{N}\right)} = \frac{\frac{1}{2}\left(1 - (-1)^k\right)}{\sin\left(\frac{1}{2}\frac{\pi k}{N}\right)}. \end{aligned}$$

Therefore, denoting $x^p = k_x + l_x$, $y^p = k_y + l_y$, $x^m = k_x - l_x$, $y^m = k_y - l_y$,

$$S_{cs,x} = S_{sc,x} = \frac{1}{4} \left[\frac{1 - (-1)^{x^p}}{\sin\left(\frac{1}{2}\frac{\pi}{N}x^p\right)} + \frac{1 - (-1)^{x^m}}{\sin\left(\frac{1}{2}\frac{\pi}{N}x^m\right)} \right]$$

$$S_{cs,y} = S_{sc,y} = \frac{1}{4} \left[\frac{1 - (-1)^{y^p}}{\sin\left(\frac{1}{2}\frac{\pi}{N}y^p\right)} + \frac{1 - (-1)^{y^m}}{\sin\left(\frac{1}{2}\frac{\pi}{N}y^m\right)} \right].$$

These terms vanish only if x^p and x^m are both even, or if y^p and y^m are both even, and are, therefore, non-zero in general, indicating the cross-terms are not orthogonal. The inner product between two basis elements of the non-separable transform is, therefore, as follows:

$$S_{cs,x}S_{cs,y} + S_{sc,x}S_{sc,y} = S_{cs,x}S_{cs,y} + S_{cs,x}S_{cs,y} = 2S_{cs,x}S_{cs,y}.$$

D Analytical solution to the modified SSF problem

The core of the unitary SSF problem is the following nonlinear equation system:

$$\begin{cases} \beta(\xi_0 - \beta\xi) + \frac{\lambda}{|c^i|}\xi = 0 \\ \beta(\eta_0 - \beta\eta) + \frac{\lambda}{|c^i|}\eta = 0, \end{cases} \quad (59)$$

where $\beta \triangleq \frac{1}{\alpha}(1 - \alpha)$ and $|c^i| = \sqrt{\xi^2 + \eta^2}$. The first equation can be written as follows:

$$0 = \beta(\xi_0 - \beta\xi) + \frac{\lambda}{|c^i|}\xi$$

$$0 = \beta\xi_0 + \xi \left(\frac{\lambda}{|c|} - \beta^2 \right)$$

$$\xi = \beta\xi_0 \frac{1}{\beta^2 - \frac{\lambda}{|c|}} = \beta\xi_0 \frac{1}{\frac{\beta^2|c| - \lambda}{|c|}}$$

$$\xi = \beta\xi_0 \frac{|c|}{\beta^2|c| - \lambda}$$

$$\eta = \beta\eta_0 \frac{|c|}{\beta^2|c| - \lambda}$$

$$\frac{\xi}{\eta} = \frac{\xi_0}{\eta_0}$$

$$\therefore \xi = \frac{\xi_0}{\eta_0}\eta \triangleq \gamma\eta.$$

Therefore,

$$|c^i| = \sqrt{\xi^2 + \eta^2} = \sqrt{\eta^2(1 + \gamma^2)} = |\eta| \sqrt{1 + \gamma^2}.$$

Further, according to (46), after substituting $|\eta| \sqrt{1 + \gamma^2}$ for $|c^i|$, we have:

$$\begin{aligned}
l &= \frac{1}{2} \left| c_0^i - \frac{1}{\alpha} (1 - \alpha) (\xi + j\eta) \right|^2 + \lambda \sqrt{1 + \gamma^2} |\eta| \\
&= \frac{1}{2} \left| c_0^i - \eta \frac{1}{\alpha} (1 - \alpha) (\gamma + 1j) \right|^2 + \lambda \sqrt{1 + \gamma^2} |\eta| \\
&= \frac{1}{2} \left| \xi_0 - \gamma \eta \frac{1}{\alpha} (1 - \alpha) + j \left(\eta_0 - \eta \frac{1}{\alpha} (1 - \alpha) \right) \right|^2 + \lambda \sqrt{1 + \gamma^2} |\eta| \\
&= \frac{1}{2} \left[(\xi_0 - \gamma \beta \eta)^2 + (\eta_0 - \eta \beta)^2 \right] + \lambda \sqrt{1 + \gamma^2} |\eta|.
\end{aligned}$$

Taking the derivative w.r.t η , we obtain:

$$\begin{aligned}
\frac{\partial l}{\partial \eta} &= -\gamma \beta (\xi_0 - \gamma \beta \eta) - \beta (\eta_0 - \eta \beta) + \lambda \sqrt{1 + \gamma^2} \text{sign} \{ \eta \} \\
&= -\beta (\eta_0 + \gamma \xi_0) + \beta^2 (1 + \gamma^2) \eta + \lambda \sqrt{1 + \gamma^2} \text{sign} \{ \eta \}.
\end{aligned}$$

If $\eta > 0$, we obtain:

$$\begin{aligned}
0 &= -\beta (\eta_0 + \gamma \xi_0) + \beta^2 (1 + \gamma^2) \eta + \lambda \sqrt{1 + \gamma^2} \\
\eta &= \frac{\beta (\eta_0 + \gamma \xi_0) - \lambda \sqrt{1 + \gamma^2}}{\beta^2 (1 + \gamma^2)},
\end{aligned}$$

with the condition for $\eta > 0$:

$$\begin{aligned}
\eta > 0 &\Leftrightarrow \frac{\beta (\eta_0 + \gamma \xi_0) - \lambda \sqrt{1 + \gamma^2}}{\beta^2 (1 + \gamma^2)} > 0 \\
&\Leftrightarrow \lambda < \frac{\beta (\eta_0 + \gamma \xi_0)}{\sqrt{1 + \gamma^2}}.
\end{aligned}$$

Similarly, for $\eta < 0$, we obtain:

$$\begin{aligned}
0 &= -\beta (\eta_0 + \gamma \xi_0) + \beta^2 (1 + \gamma^2) \eta - \lambda \sqrt{1 + \gamma^2} \\
\eta &= \frac{\beta (\eta_0 + \gamma \xi_0) + \lambda \sqrt{1 + \gamma^2}}{\beta^2 (1 + \gamma^2)}, \\
\eta < 0 &\Leftrightarrow \frac{\beta (\eta_0 + \gamma \xi_0)}{\sqrt{1 + \gamma^2}} < -\lambda.
\end{aligned}$$

Therefore, the optimal solution for η is:

$$\eta = \begin{cases} \frac{\beta z_0 - \lambda \gamma_0}{\beta^2 \gamma_0^2}, & \frac{\beta z_0}{\gamma_0} > \lambda \\ 0, & \left| \frac{\beta z_0}{\gamma_0} \right| \leq \lambda \\ \frac{\beta z_0 + \lambda \gamma_0}{\beta^2 \gamma_0^2}, & -\lambda > \frac{\beta z_0}{\gamma_0}, \end{cases} \quad (60)$$

where $z_0 = \eta_0 + \gamma \xi_0$ and $\gamma_0 = \sqrt{1 + \gamma^2}$.

E MMSE-based decomposition

Recalling (31), the signal $c(x, y)$ can be represented in matrix form as $D_\theta c$. Therefore, the signal $g(x, y)$ is given in its vectorized version by

$$g = \alpha \Phi \xi + (1 - \alpha) D_\theta c \quad (61)$$

$$= \begin{bmatrix} \alpha \Phi & 0 \\ 0 & (1 - \alpha) D_\theta \end{bmatrix} \begin{bmatrix} \xi \\ c \end{bmatrix} \triangleq Mv. \quad (62)$$

Let us summarize the variables to be estimated from (62): α is the mixing parameter; Φ is a variable which depends on the Hurst parameter, H ; $\{m_i, \theta_i\}_{i=1}^K$ and $\{u_i, v_i\}_{i=1}^K$ are the magnitude-phase pairs and the frequency locations from (29), respectively. Denoting the parameter set as Θ , its maximum-likelihood estimator, $\hat{\Theta}(g)$ is given by:

$$\hat{\Theta}(g) = \arg \max \mathcal{L}(\Theta|g) := p(g|\Theta),$$

where $p(g|\Theta)$ is the density of g conditioned on the parameter set, Θ . We observe that this problem is intractable under our current assumptions; whereas the first term in (62) is Gaussian, the second term is a linear combination of log-normally-distributed variables, which does not lend itself to a simple analytic expression. We, therefore, solve this problem assuming that c are normally distributed. Given Θ ,

$$g = Mv \sim \mathcal{N}(\underline{\mu}, \Sigma),$$

where

$$\begin{aligned} \underline{\mu} &= \mathbb{E}\{g\} = \alpha \Phi \mathbb{E}\{\xi\} + (1 - \alpha) D_\theta \mathbb{E}\{c\} \\ &= (1 - \alpha) D_\theta \underline{\mu}_c, \end{aligned}$$

and

$$\begin{aligned}\Sigma &= \text{Cov}\{Mv\} = M\text{Cov}\{v\} M^T \\ &= M\Sigma_V M^T.\end{aligned}$$

Σ_V is a diagonal matrix, whose first K entries are equal to one, and the rest of the entries correspond to c and are denoted $\underline{\sigma}_c^2$. Therefore,

$$\Sigma = \begin{bmatrix} \alpha^2\Phi\Phi^T & 0 \\ 0 & (1-\alpha)^2 D_\theta \underline{\sigma}_c^2 D_\theta^T \end{bmatrix} = \begin{bmatrix} \alpha^2\Sigma_H & 0 \\ 0 & (1-\alpha)^2 D_\theta \underline{\sigma}_c^2 D_\theta^T \end{bmatrix},$$

where Σ_H is an fBm/fGn covariance matrix. The likelihood function then follows a multivariate normal distribution,

$$\mathcal{L}(\Theta|g) = \frac{1}{((2\pi)^n |\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1}(\underline{x} - \underline{\mu})\right\}.$$

The maximum-likelihood estimator for Σ is given by $\hat{\Sigma} = \frac{1}{n} \sum_i (\underline{x}_i - \underline{\mu})^T (\underline{x}_i - \underline{\mu})$. However, we would like to evaluate the derivative

$$\frac{\partial \mathcal{L}(\Theta|g)}{\partial \theta_i} = \text{tr}\left(\frac{\partial \mathcal{L}(\Theta|g)}{\partial \Sigma} \cdot \frac{\partial \Sigma}{\partial \theta_i}\right).$$

Unfortunately, this gives rise to far more complex equations than can be solved precisely.