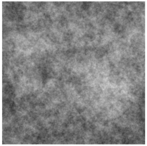
Local estimation of the local roughness of non-homogenous Brownian textures by convolution neural networks.

Frédéric RICHARD

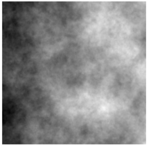
Institute of Mathematics of Marseille, Aix-Marseille University.

Stochastic Geometry Days
June, 12-16, 2023.
Dijon, France.

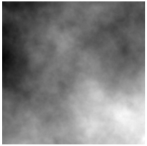
Global regularity of textures



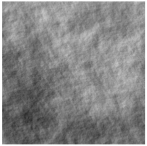
$H = 0.1$



$H = 0.5$

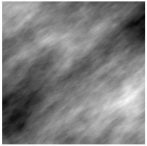


$H = 0.8$



$H = 0.1$

$H = 0.1$

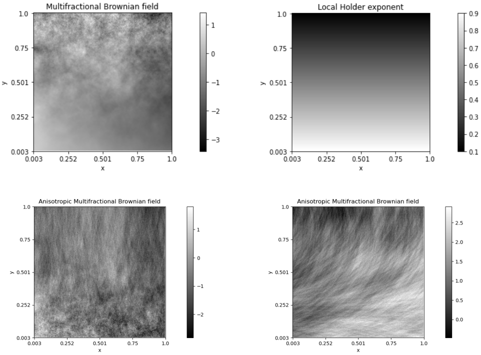


$H = 0.5$

Hölder exponent H of a field Z: for any compact set C,

$$H \underset{\text{a.s.}}{=} \sup \left\{ \alpha, \sup_{y \neq y' \in C} \frac{|Z(y) - Z(y')|}{|y - y'|^{\alpha}} < +\infty \right\}$$

Local regularity of textures



Local Hölder exponent H_x of a field Z at position x

$$H_x \underset{\text{a.s.}}{=} \sup \left\{ \alpha, \exists \rho > 0, \sup_{y \neq y' \in B(x,\rho)} \frac{|Z(y) - Z(y')|}{|y - y'|^{\alpha}} < +\infty \right\}$$

Anisotropic fractional Brownian field

Gaussian field with stationary increments defined by

$$Y_{\tau,\beta}(x) = \int_{\mathbb{R}^2} \left(e^{i\langle \omega, x\rangle} - 1\right) \sqrt{\tau(\arg \omega)} \|\omega\|^{-\beta(\arg \omega)-1} d\widehat{W}(\omega),$$

with two directional functions τ and β (topothesy and Hurst functions).

Local Hölder regularity given, at any position x, by the Hurst index :

$$H = \operatorname{ess\,inf}_{s} \{\beta(s), \tau(s) > 0\}.$$

Estimations methods:

- Periodogram, quadratic variations, wavelets,...
- Asymptotic normality of estimators.

[ref. Bonami and Estrade, 2003]

Multi-fractional anisotropic fractional Brownian field

Gaussian field defined by

$$\tilde{Y}_{\tau,\beta}(x) = \int_{\mathbb{R}^2} \left(e^{i\langle\omega,x\rangle} - 1\right) \sqrt{\tau_x(\arg\omega)}\|\omega\|^{-\beta_x(\arg\omega)-1} d\widehat{W}(\omega),$$

where W is a complex Brownian measure and, τ_x and β_x, two spatially varying functions (topothesy and Hurst functions).

Let

$$H_x = \operatorname{ess\,inf}_s \{\beta_x(s), \tau_x(s) > 0\},$$

and

$$\tilde{\tau}_x(s) = \tau_x(s)\mathbf{1}_{\beta_x(s)=H_x}.$$

Then, $\tilde{Y}_{\tau,\beta}$ is tangent at x (l.a.s.s.) to an AFBF of topothesy and Hurst functions $\tilde{\tau}_x$ and H_x.

Hölder regularity at x given by H_x.

[ref. Benassi et al, 97; Polisano et al, 2014; Vu and R., 2020]

Estimation of the local regularity.

Previous works:

- Quadratic variations: Coeurjoly, 2001; Vu et R., 2020.
- Wavelet leaders coupled with a regularization by total variation: Pascal, Pustelnik, Abry, 2021.

Main numerical challenges:

- achieve a good spatial precision,
- be robust to image noise and transforms (e.g. encoding of image values),
- develop benchmarks.

Local analysis of images.

- Let Z be observed on a grid: $Z^N[m] = Z(\frac{m}{N}), m \in [\![1, N]\!]^2$.
- Given some $u_{jk} = \rho_{jk}(\cos\varphi_j, \sin\varphi_j) \in \mathbb{Z}^2 \backslash \{(0,0)\}$, rescale the image of a factor ρ_{jk} and rotate it of an angle φ_k

$$T_{jk} = \rho_{jk} \begin{pmatrix} \cos(\varphi_j) & -\sin(\varphi_j) \\ \sin(\varphi_j) & \cos(\varphi_j) \end{pmatrix}.$$

- Convolve the transformed images

$$V_{jk}^N[m] = \sum_n v[n] Z^N[m - T_{jk} n]$$

  with a kernel v annihilating polynomials of order < 2.
- Compute the quadratic variations in neighborhood of some positions x_i:

$$W_{ijk}^N = \frac{1}{|\mathcal{V}_N|} \sum_{m \in \mathcal{V}_N} (V_{jk}^N[m + p_i])^2.$$

Estimation of the local Hurst index

Theorem

Let $Y_{ijk}^N = \log(W_{ijk}^N)$ and $x_{jk0} = \log(\rho_{jk}^2)$. Then, under appropriate assumptions,

$$N^{\frac{d}{2}}(Y^N - \zeta^N) \xrightarrow[N \to +\infty]{d} \mathcal{N}(0, \Sigma),$$

for a covariance matrix Σ, and an expectation ζ^N of the form

$$\zeta_{ijk}^N = x_{jk0} H_{x_i} + \beta_{ij}^N.$$

Estimation: For any i, let $\theta_{(i)} = (H_{x_i}, \beta_{i1}^N, \cdots, \beta_{iJ}^N)^T$, then

$$\zeta_{(i)}^N = X\theta_{(i)} + \epsilon, \text{with } \epsilon \sim \mathcal{N}(0, \Sigma_{(i)})$$

So that

$$\widehat{H}_{x_i} = (1, 0, \cdots, 0)(X^T \Sigma_{(i)}^{-1} X)^{-1} X^T \Sigma_{(i)}^{-1} Y_i^N$$

[Ref. Hu and Richard, 2020]

Analogy with neural networks

- Construction of a feature vector:
  - Convolution layer:

$$V_{jk}^N[m] = \sum_n v[n] Z^N[m - T_{jk} n] = \sum_n v_{jk}[k] Z^N[m - n].$$

  - Square activation: $(V_{jk}^N[m])^2$
  - Average pooling:

$$W_{ijk}^N = \frac{1}{|\mathcal{V}_N|} \sum_{m \in \mathcal{V}_N} (V_{jk}^N[m + p_i])^2.$$

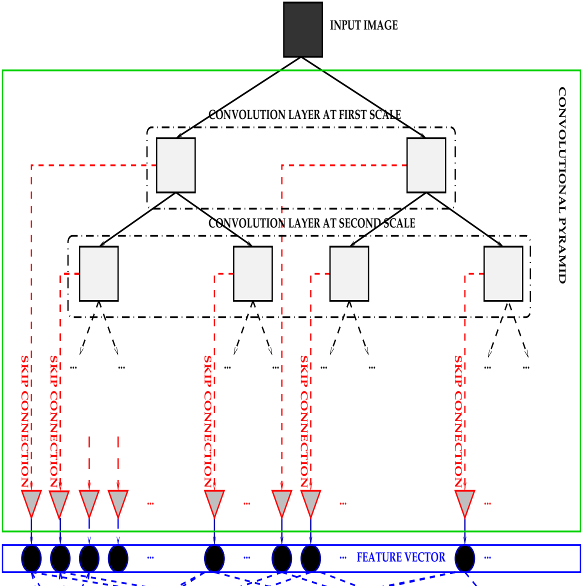
  - Log activation:

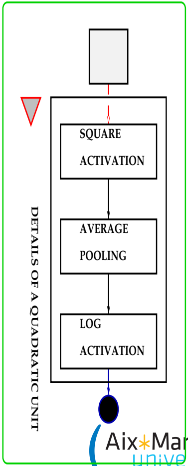
$$Y_{ijk}^N = \log(W_{ijk}^N)$$

- Regression with a dense layer:

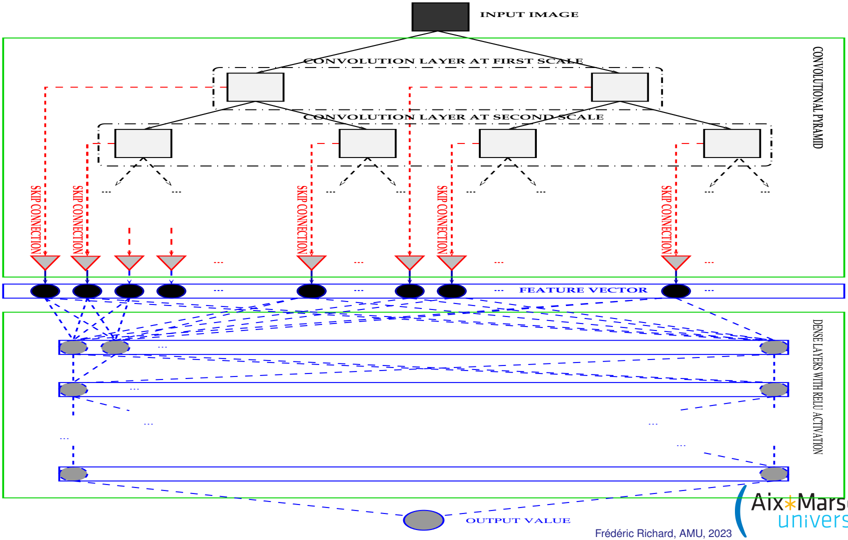
$$\widehat{H}_{x_i} = (1, 0, \cdots, 0)(X^T \Sigma_{(i)}^{-1} X)^{-1} X^T \Sigma_{(i)}^{-1} Y_i^N.$$

Design of the CNN: convolutional part





Design of the CNN: complete architecture



CNN Learning

- Parameter summary:

Layer type	height	size	parameters
Conv pyramid	5	$3 \times 3 + 1$	6 138
Dense layers	8	20 + 1	5 501
Total			11 639

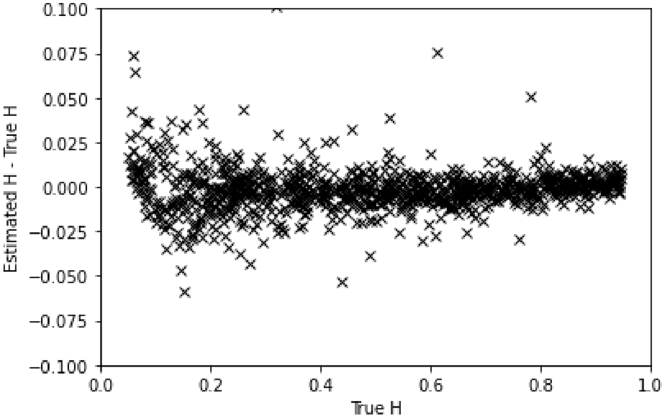
- Generation of a dataset using the package PyAFBF:
  - Images of size 64×64 sampled from AFBF.
  - Parameters of AFBF models are set randomly; the Hurst index is uniformly sampled.
- Set sizes

	Size
Training set	98 000
Validation set	1 000
Test set	1 000
Total	100 000

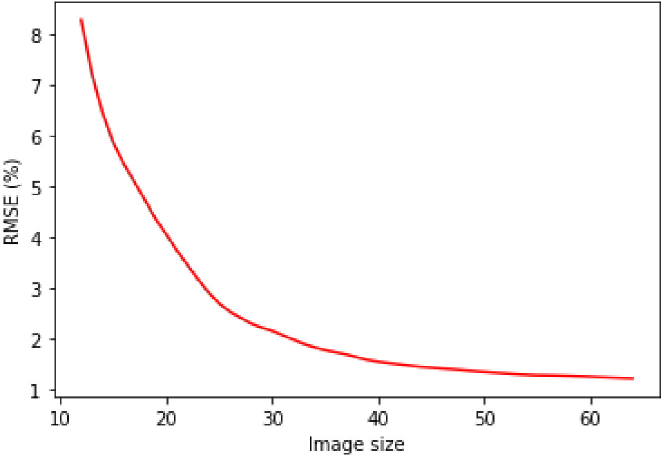
- Tensorflow (Keras): optimizer (Adam), loss (MSE), batch size (20) , nb epochs (20), time (46 min).

Error analysis

Test error: **1.23 %** (on 1000 images of size 64 $\times$ 64).
to be compared with 3.5 % for the classical method.
Criterium: root mean square error (RMSE, in percent).



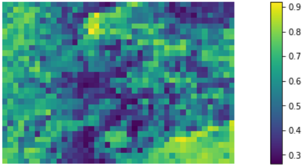
Error as a function of the image size.



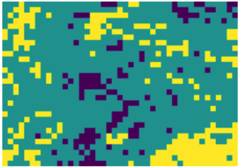
Rmk: a gain of 4 % w.r.t. the classical method.

Application to image segmentation

Image



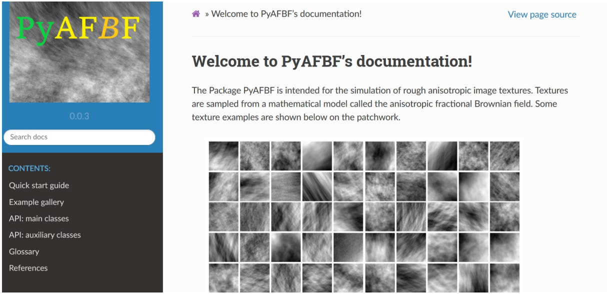
Hurst index estimates



Segmentation

- Source: Max Planck Institute for Meteorology (Understanding Clouds from Satellite Images).
- Processing :
  0.48s per image (1200×1750) with non-overlapping patches (40×40).

PyAFBF



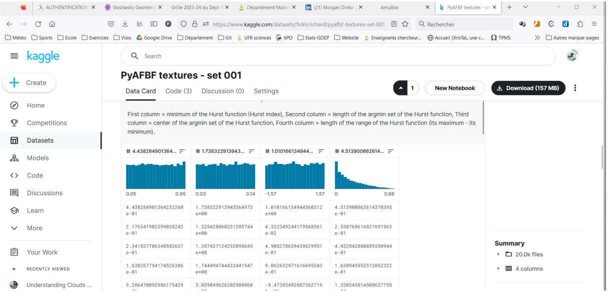
Features:

- Simulation of rough anisotropic textures,
- Computation of field features and texture attributes,
- Random definition of simulated fields,
- Extensions to related fields.

Complementary packages:

- PyAFBFdb: to build database for reproducible research.
- PyAFBFest (soon): to estimate model parameters.

Benchmark on Kaggle.



- Repository of 20000 images with associated features.
- A code to reproduce this work.
- Features to estimates from the Hurst function (β):
  - Hurst index: $H = \min_s \beta(s)$.
  - Length of the argmin set: $\mathcal{L}(\arg\min_s \beta(s))$.
  - Center of the argmin set: $\mathcal{C}(\arg\min_s \beta(s))$.
  - Range length: $\mathcal{R} = \max_s \beta(s) - H$
- Features H, $\mathcal{L}$, $\mathcal{C}$ are uniformly distributed over the set.