PyAFBF: a Python library for sampling image textures from the anisotropic fractional Brownian field.

Frédéric J. P. Richard

1 Aix Marseille University, CNRS, Centrale Marseille, I2M, UMR 7373, Marseille, France.

Summary

The Python library PyAFBF is devoted to the simulation of anisotropic textures of image. These textures are sampled from a mathematical model called the anisotropic fractional Brownian field (AFBF) (Bonami & Estrade, 2003); see Figure 1 for an illustration. The library offers several features. Users can generate a wide variety of textures by setting the simulated model manually or randomly either manually or randomly. It also includes tools to compute model features (regularity, anisotropy,...) which may serve as attributes to describe generated textures. The library further offers the possibility to sample heterogeneous textures from random field models related to the AFBF.
Statement of need

For the simulation of random fields, the library PyAFBF relies upon the turning-band method developed in (Biermé et al., 2015). This method was historically designed to facilitate research on the anisotropic fractional Brownian field (AFBF) (Bonami & Estrade, 2003) and related models (Benassi et al., 1997; Guyon & Perrin, 2000; Peltier & Levy Vehel, 1996; Polisano et al., 2014; Vu & Richard, 2020). The library is of interest for researchers in image processing where random fields can serve as texture or noise models for medical images (Biermé et al., 2009; Biermé & Richard, 2011; Richard, 2015, 2016a; Richard & Biermé, 2010) or photographic films (Richard, 2017). It could also be interesting for machine learning researchers who could include the random field simulation in the design or the learning of image generative models such as Generative Adversarial Networks.

Besides, AFBF are characterized by two parameters called the topothesy function and the Hurst functions; see below for a definition. These parameters, as well as derived features, are important to characterize directional properties of the field and describe the image texture anisotropy. They can be used to classify or segment images, and detect image abnormalities. Achieving such tasks requires estimating parameters and features. However, the statistical estimation of AFBF features is still a topic of research. In particular, actual methods for the estimation of the topothesy and Hurst functions are incomplete: the method proposed in

(Richard, 2017) only estimates the topothesy function in directions where the Hurst function is minimal. The method in (Biermé & Richard, 2008) can only be used for the estimation of the Hurst function in the horizontal and vertical directions.

In this research context, simulations of AFBF have been particularly useful for the evaluation of estimation methods (Biermé & Richard, 2008; Richard, 2017, 2015, 2016b, 2016a; Richard & Biermé, 2010). Hence, in the future, the library PyAFBF could be used by mathematicians to design experiments and assess method performances. In particular, the library could be the basis for collaborative works aiming at the development of benchmarks and data challenges concerning the estimation issue.

The simulation of random fields is a classical topic of spatial statistics (Chilès & Delfiner, 2012; Cressie, 2015; Lantuéjoul, 2013). As reviewed in (Liu et al., 2019), there are some R packages devoted to this topic, among which are:

- RandomFields (Schlather et al., 2015) that enables the simulation of stationary fields and also some non-stationary Gaussian random fields such max-stable fields
- FieldSim (Brouste et al., 2007) that allows the simulation of manifold indexed Gaussian field

None of these package deal directly with anisotropic fractional Brownian fields. The package FieldSim deals with mono- and multi- fractional Brownian fields but only in an isotropic setting. The package RandomFields offers a wide range of methods to simulate stationary and non-stationary, isotropic and anisotropic random fields. However, it only handles geometric and zonal anisotropies, which both differ from the anisotropy of an AFBF. Moreover, it is not specifically devoted to models derived from the fractional Brownian fields. Hence, the package PyAFBF is complementary to these R packages. In Python, the implementations of random field simulation methods are less developed. It is the purpose of the package python-randomfields and dune-randomfield. It is also a part of packages spam and dorie. But, none of these package enables the simulation of AFBF. Hence, the package PyAFBF offers original simulation tools to a large community of Python developers.

**Definition and simulation of an AFBF.**

An AFBF $Z$ is a Gaussian non-stationary random field with stationary increments whose semi-variograms are of a form

$$v(h) = \frac{1}{2} \mathbb{E}((Z(x+h) - Z(x))^2) = \frac{1}{2} \int_{-\pi/2}^{\pi/2} \tau(\theta) |\langle h, u(\theta) \rangle|^{2\beta(\theta)} \, d\theta,$$

which is characterized by two $\pi$-periodic functions $\tau$ and $\beta$ called the topothesy function and the Hurst function, respectively. These functions determine the properties of the AFBF and the textures that are sampled from it.

The package PyAFBF proposes some convenient representations for these functions (Fourier, step functions,…) that enable users to easily set an AFBF, either manually or at random.

Using the package PyAFBF, image textures are realizations of an AFBF on a discrete grid. AFBF are simulated using a turning band fields (Biermé et al., 2015). These fields are defined, for some set of angles $\langle \varphi_k, k = 1, \cdots, K \rangle$ in $[-\pi/2, \pi/2]$ and of appropriate non-negative weights $\langle \lambda_k, k = 1, \cdots, K \rangle$, as

$$Z_\varphi(x) = \sum_{k=1}^{K} \lambda_k X_k(\langle u(\varphi_k), x \rangle),$$

where $X_k$ are independent Brownian motions with Hurst index $h_k$. The package includes a Python class to handle turning-band fields, simulate them and compute their properties.

### Availability and Community Guidelines

The package **PyAFBF** can be downloaded from the Github repository. Documentation, which includes a quickstart guide, a gallery of examples and API, is available at the PyAFBF site. Users and contributors are welcome to contribute, request features, and report bugs via Github.

### References


