PATCH REDUNDANCY IN IMAGES: A STATISTICAL TESTING FRAMEWORK AND SOME APPLICATIONS
Valentin de Bortoli, Agnès Desolneux, Bruno Galerne, Arthur Leclaire

To cite this version:
Valentin de Bortoli, Agnès Desolneux, Bruno Galerne, Arthur Leclaire. PATCH REDUNDANCY IN IMAGES: A STATISTICAL TESTING FRAMEWORK AND SOME APPLICATIONS. SIAM Journal on Imaging Sciences, Society for Industrial and Applied Mathematics, 2019, 12 (2), pp.893-926. hal-01931733v2

HAL Id: hal-01931733
https://hal.archives-ouvertes.fr/hal-01931733v2
Submitted on 30 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Patch redundancy in images: a statistical testing framework and some applications *

Valentin De Bortoli†, Agnès Desolneux‡, Bruno Galerne‡, and Arthur Leclaire§

Abstract. In this work we introduce a statistical framework in order to analyze the spatial redundancy in natural images. This notion of spatial redundancy must be defined locally. To do so, we define an a-similarity function which, given one image, computes a dissimilarity measurement between patches. To derive a criterion for taking a decision on the similarity between two patches we present an a contrario model. Namely, two patches are said to be similar if the associated dissimilarity measurement is unlikely to happen in a background model. Choosing Gaussian random fields as background models we derive non-asymptotic expressions for the probability distribution function of similarity measurements. We present an algorithm in order to assess redundancy in natural images and discuss applications in denoising, periodicity analysis and texture ranking.

Key words. patch, redundancy, statistical framework, a contrario method, image denoising, texture, periodicity analysis.

AMS subject classifications. 60F05, 60F15, 60G15, 60G60, 62H15, 62H35, 65S05, 65T50

1. Introduction. In many image processing applications, using local information combined with the knowledge of long-range spatial arrangement is crucial. The spatial redundancy on sub-images called patches, encodes the small scale structure of the image as well as its large scale organization. More precisely, local information is encoded in the patch content and the large scale organization is contained in the redundancy of this information across the patches of the image. For example, patch-based inpainting techniques, such as [10, 33], assign patches of a known region to patches of an unknown region. Namely, each patch position on the border of the unknown region is associated to an offset corresponding to the best patch according to the partial available information. In [33] the authors replace the search on the whole image by a search among the most redundant offsets in the known region. This allows the authors of [33] to retrieve long-range spatial structure in the unknown part of the image. Another famous application of spatial redundancy can be found in denoising, with the seminal work (Non-Local means) of Buades and coauthors [5], in which the authors propose to replace a noisy patch by the mean over all spatially redundant patches.

Last but not least, spatial redundancy is of crucial importance in exemplar-based texture synthesis. In this paper we define textures as images containing repeated patterns but also reflecting randomness in the arrangement of these patterns. Among textures, one important class is given by the microtextures in which no individual object can be clearly delimited. In the periodic case, a more precise definition will be given in Definition 4. These microtexture models can be described by Gaussian random fields [62, 27, 42, 68]. Parametric models using features such as wavelet transform coefficients [55], scattering transform coefficients [59] or

*Submitted to the editors DATE.
†CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France
‡Institut Denis Poisson, Université d’Orléans, Université de Tours, CNRS
§Univ. Bordeaux, IMB, Bordeaux INP, CNRS, UMR 5251, F-33400 Talence, France.
convolutional neural network outputs [29] have been proposed in order to derive image models with more structure. On the other hand, non-parametric patch-based algorithms such as [25, 24, 38, 56, 28] propose to use most similar patches in order to fill the new texture images, similarly to inpainting techniques.

All these techniques lift images in spaces with dimensions higher than the original image space, and make use of the redundancy of the lifting to extract important structural information. There exist two main types of lifting: feature extraction or patch extraction. Feature extraction relies on the use of filters, linear or non-linear, which aim at selecting substantial local information. Among popular kernels are oriented and multiscale filters, which happened to be identified as early processing in mammal vision systems [13, 35]. These last years have seen the rise of neural networks in which the filter dictionary is no longer given as an input but learned through a data-driven optimization procedure [60]. On the other hand, patch-based methods rely on the assumption that image processing tasks are simplified when conducted in the higher dimensional patch space.

Every analysis performed in a lifted space, built via feature extraction or patch extraction, relies on the comparison of points in this space. In patch-based lifted spaces, we aim at finding dissimilarity functions such that two patches are visually close if the dissimilarity measurement between them is small. In this paper we focus on the square Euclidean distance but other choices could be considered [65, 64, 15, 17].

This leads us to consider a statistical hypothesis testing framework to assess similarity (or dissimilarity) between patches. The null hypothesis is defined as the absence of local structural similarities in the image. Reciprocally the alternative hypothesis is defined as the presence of such similarities. There exists a wide variety of tractable models exhibiting no similarity at long-range, like Gaussian random fields [62, 27, 42, 68] or spatial Markov random fields [11], whereas sampling and inference in very structured models rely on optimization procedures and may be computationally expensive, their distribution being the limit of some Markov chain [70, 47] or some stochastic optimization procedure [4]. This encourages us to consider an a contrario approach, i.e. we do not consider the alternative hypothesis and focus on rejecting the null hypothesis. This framework was successfully applied in many areas of image processing [14, 19, 20, 1, 7] and aims at identifying structure events in images. This statistical model takes its roots in the fundamental work of the Gestalt theory [21]. One of its principle, the non-accidentalness principle [46] or Helmholtz principle [69, 20], states that no structure is perceived in a noise model. To be precise, in our case of interest, we want to assess that no spatial redundancy is perceived in microtexture models. This methodology allows us to only design a locally structured background model to define a null hypothesis. Combining a contrario principles and patch-based measures, we propose an algorithm to identify auto-similarities in images.

We then turn to the implementation of such an algorithm and illustrate the diversity of its possible applications with three examples: denoising, lattice extraction, and periodicity ranking of textures. In our denoising application we propose a modification of the celebrated Non-Local means algorithm [5] (NL-means) by inserting a threshold in the selection of similar patches. Using an a contrario model we are able to give probabilistic control on the patch reconstruction.

We then focus on periodicity detection and, more precisely, lattice extraction. Periodicity
in images was described as an important feature in early mathematical vision [32]. Most of the proposed methods to analyze periodicity rely on global measurements such as the modulus of the Fourier transform [49] or the autocorrelation [43]. These global techniques are widely used in crystallography where lattice properties, such as the angle between basis vectors, are fundamental [50, 58]. Since all of our measurements are local, we are able to identify periodic similarities even in images which are not periodic but present periodic parts, for instance if two crystal structures are present in a single crystallography image. We draw a link between the introduced notion of auto-similarity and the inertia measurement in co-occurrence matrices [32]. We then introduce our lattice proposal algorithm which combines a detection map, *i.e.* the output of our redundancy detection algorithm, and graphical model techniques, as in [53], in order to extract lattice basis vectors.

Our last application concerns texture ranking. Since the definition of texture is broad and covers a wide range of images, it is a natural question to identify criteria in order to distinguish textures. In [45], the authors use a classical measure for distinguishing textures: regularity. In this work, we narrow this criterion and restrict ourselves to the study of periodicity in texture images. The proposed graphical model inference naturally gives a quantitative measurement for texture periodicity ranking. We give an example of ranking on 25 images of the Brodatz set.

Our paper is organized as follows. An *a contrario* framework for local similarity detection is proposed in Section 2. In the *a contrario* framework, a background model, corresponding to the null hypothesis, is required. The consequence of choosing Gaussian models as background models is investigated and a redundancy detection algorithm is proposed in Section 3. The rest of the paper is dedicated to some examples of application of the proposed framework. After reviewing one of the most popular method in image denoising we introduce a denoising algorithm in Section 4.1 and present our experimental results in Section 4.2. Local dissimilarity measurements can be used as periodicity detectors. The link between the locality of the introduced functions and the literature on periodicity detection problems is investigated in Section 5.1. An algorithm for detecting lattices in images is given in Section 5.2 and numerical results are presented in Section 5.3. In our last experiment in Section 5.4, we introduce a criterion for measuring texture periodicity. We conclude our study and discuss future work in Section 6.

2. An *a contrario* framework for auto-similarity. We first introduce a notion of dissimilarity between patches of an input image.

**Definition 1 (Auto-similarity).** Let $u$ be an image defined over a domain $\Omega = [0, M - 1]^2 \subset \mathbb{Z}^2$, with $M \in \mathbb{N}\setminus\{0\}$. Let $\omega \subset \mathbb{Z}^2$ be a patch domain. We introduce $P_{\omega}(u) = (\hat{u}(y))_{y \in \omega}$ the patch at position $\omega$ in the periodic extension of $u$ to $\mathbb{Z}^2$, denoted by $\hat{u}$. We define the auto-similarity with patch domain $\omega$ and offset $t \in \mathbb{Z}^2$ by

$$\text{AS}(u, t, \omega) = \|P_{t+\omega}(u) - P_{\omega}(u)\|_2^2.$$ 

The auto-similarity computes the distance between a patch of $u$ defined on a domain $\omega$ and the patch of $u$ defined by the domain $\omega$ shifted by the offset vector $t$. In what follows, we
introduce an *a contrario* framework on the auto-similarity. This framework will allow us to derive an algorithm for detecting spatial redundancy in natural images. In this section we fix an image domain \( \Omega \subseteq \mathbb{Z}^2 \) and a patch domain \( \omega \subseteq \Omega \). We recall that our final aim is to design a criterion that will answer the following question: are two given patches similar? This criterion will be given by the comparison between the value of a dissimilarity function and a threshold \( a \). We will define the threshold \( a \) so that few similarities are identified in the null hypothesis model, *i.e.* similarity does not occur “just by chance”. Thus we can reformulate the initial question: is the similarity output of a dissimilarity function between two patches small enough? Or, to be more precise, how can we set the threshold \( a \) in order to obtain a criterion for assessing similarity between patches?

This formulation agrees with the *a contrario* framework [21] which states that geometrical and/or perceptual structure in an image is meaningful if it is a rare event in a background model. This general principle is sometimes called the Helmholtz principle [69] or the non-accidentalness principle [46]. Therefore, in order to control the number of similarities identified in the background model, we study the probability density function of the auto-similarity function with input random image \( U \) over \( \Omega \). We will denote by \( P_0 \) the probability distribution of \( U \) over \( \mathbb{R}^\Omega \), the images over \( \Omega \). We will assume that \( P_0 \) is a microtexture model, see Definition 4 below for a precise definition of such a model. We define the following significant event which encodes spatial redundancy: 
\[
\text{AS}(u, t, \omega) \leq a(t),
\]
where \( a \), the threshold function, is defined over the offsets \( t \in \mathbb{Z}^2 \) but also depends on other parameters such as \( \omega \) or \( P_0 \). The dependency of \( a \) with respect to \( t \) cannot be omitted. For instance, even in a Gaussian white noise \( W \), the probability distribution function of \( \text{AS}(W, t, \omega) \) depends on \( t \).

The Number of False Alarms (NFA) is a crucial quantity in the *a contrario* methodology. A false alarm is defined as an occurrence of the significant event in the background model \( P_0 \). We recall that in our model the significant event is patch redundancy. This test must be conducted for every possible configurations of the significant event, *i.e.* in our case we test every possible offset \( t \). The NFA is then defined as the expectation of the number of false alarms over all possible configurations. Bounding the NFA ensures that the probability of identifying \( k \) offsets with spatial redundancy is also bounded, see Proposition 1. In what follows we give the definition of the NFA in the spatial redundancy context.

**Definition 2 (NFA).** Let \( U \sim P_0 \), where \( P_0 \) is a background microtexture model. We define the auto-similarity probability map \( \text{AP} \) for any \( t \in \Omega, \omega \subseteq \Omega \) and \( a \in \mathbb{R}^\Omega \) by

\[
\text{AP}(t, \omega, a) = P_0 [\text{AS}(U, t, \omega) \leq a(t)].
\]

We define the auto-similarity expected number of false alarms \( \text{ANFA} \) by

\[
\text{ANFA}(\omega, a) = \sum_{t \in \Omega} \text{AP}(t, \omega, a).
\]

Note that \( \text{AP}(t, \omega, a) \) corresponds to the probability that \( \omega + t \) is similar to \( \omega \) in the background model \( U \). For any \( t \in \Omega \), the cumulative distribution function of the auto-similarity random variable \( \text{AS}(U, t, \omega) \) under \( P_0 \) evaluated at value \( \alpha(t) \) is given by \( \text{AP}(t, \omega, \alpha(t)) \). We denote by \( q \mapsto \text{AP}^{-1}(t, \omega, q) \) the inverse cumulative distribution function, potentially defined by a
generalized inverse \((\text{AP}^{-1}(t, \omega, q) = \inf\{\alpha(t) \in \mathbb{R}, \text{AP}(t, \omega, \alpha(t)) \geq q\})\), of the auto-similarity random variable for a fixed offset \(t\), with \(q \in (0, 1)\) a quantile. We now have all the tools to control the number of detected offsets in the background model.

**Definition 3 (Detected offset).** Let \(u \in \mathbb{R}^\Omega\) be an image, \(\omega \subset \Omega\) a patch domain, and \(a \in \mathbb{R}^\Omega\). An offset \(t\) is said to be detected with respect to \(a\), if \(\text{AS}(u, t, \omega) \leq a(t)\).

Note that a detected offset in \(U \sim \mathbb{P}_0\) corresponds to a false alarm in the a contrario model. In what follows we suppose that the cumulative distribution function of \(\text{AS}(U, t, \omega)\) is invertible for every \(t \in \Omega\). This ensures that for any \(t \in \Omega\) and \(q \in (0, 1)\) we have

\[
\text{AP} \left( t, \omega, \text{AP}^{-1} (t, \omega, q) \right) = q .
\]

**Proposition 1.** Let \(\text{NFA}_{\text{max}} \geq 0\) and for all \(t \in \Omega\) define \(a(t) = \text{AP}^{-1}(t, \omega, \text{NFA}_{\text{max}} / |\Omega|)\). We have that for any \(n \in \mathbb{N}\setminus\{0\}\),

\[
\text{ANFA}(\omega, a) = \text{NFA}_{\text{max}} \quad \text{and} \quad \mathbb{P}_0 \left[ \text{"at least } n \text{ offsets are detected in } U \" \right] \leq \frac{\text{NFA}_{\text{max}}}{n} .
\]

**Proof.** Using (2.3), and \(a(t) = \text{AP}^{-1}(t, \omega, \text{NFA}_{\text{max}} / |\Omega|)\), we get

\[
\text{ANFA}(\omega, a) = \sum_{t \in \Omega} \text{AP}(t, \omega, a) = \sum_{t \in \Omega} \text{AP} \left( t, \omega, \text{AP}^{-1} (t, \omega, \text{NFA}_{\text{max}} / |\Omega|) \right) = \text{NFA}_{\text{max}} ,
\]

where the last equality is obtained using (2.4). Concerning the upper-bound, we have, using the Markov inequality and (2.2), for any \(n \in \mathbb{N}\setminus\{0\}\)

\[
\mathbb{P}_0 \left[ \text{"at least } n \text{ offsets are detected in } U \" \right] = \mathbb{P}_0 \left[ \sum_{t \in \Omega} \mathbb{I}_{\text{AS}(U, t, \omega) \leq a(t)} \geq n \right] \\
\leq \sum_{t \in \Omega} \mathbb{E} \left[ \mathbb{I}_{\text{AS}(U, t, \omega) \leq a(t)} \right] \leq \frac{\text{NFA}_{\text{max}}}{n} ,
\]

where \(\mathbb{I}_{\text{AS}(U, t, \omega) \leq a(t)} = 1\) if \(\text{AS}(U, t, \omega) \leq a(t)\) and 0 otherwise.

Thus, setting \(a\) as in Proposition 1, we have that an offset \(t \in \Omega\) is detected for an image \(u \in \mathbb{R}^\Omega\) if

\[
\text{AS}(u, t, \omega) \leq \text{AP}^{-1}(t, \omega, \text{NFA}_{\text{max}} / |\Omega|) .
\]

This a contrario detection framework can then be simply rewritten as 1) computing the auto-similarity function with input image \(u\), 2) thresholding the obtained dissimilarity map with the inverse cumulative distribution function of the computed dissimilarity function under \(\mathbb{P}_0\). The computed threshold depends on the offset and Proposition 1 ensures probabilistic guarantees on the expected number of detections under \(\mathbb{P}_0\). Using the inverse property of the inverse cumulative distribution function and (2.5), we obtain that an offset is detected if and only if

\[
\mathbb{P}_0 \left[ \text{AS}(U, t, \omega) \leq \text{AS}(u, t, \omega) \right] = \text{AP} \left( t, \omega, \text{AS}(u, t, \omega) \right) \leq \text{NFA}_{\text{max}} / |\Omega| .
\]

Therefore, the thresholding operation can be conducted either on \(\text{AS}(u, t, \omega)\), see (2.5), or on \(\text{AP} \left( t, \omega, \text{AS}(u, t, \omega) \right)\), see (2.6). This property will be used in Section 3.2 to define a similarity detection algorithm based on the evaluation of \(\text{AS}(u, t, \omega)\).
3. Gaussian model and detection algorithm.

3.1. Choice of background model. In this section we compute $AP(t, \omega, \alpha)$, i.e. the cumulative distribution function of the similarity function under the null hypothesis model, with a Gaussian background model. Indeed, if the background model is simply a Gaussian white noise the similarities identified by the a contrario algorithm are the ones that are not likely to be present in the Gaussian white noise image model. More generally, we consider stationary Gaussian random fields defined in the following way: we introduce an image $f \in \mathbb{R}^\Omega$ which contains the microtexture information we want to discard in our a contrario model. In what follows we give the definition of the microtexture model associated to $f$.

Definition 4 (Microtexture model). Let $f \in \mathbb{R}^\Omega$, we define the associated microtexture model $U$ by setting, $U = f \ast W$, where $\ast$ is the periodic convolution operator over $\Omega$ given by $v \ast w(x) = \sum_{y \in \Omega} v(y) w(x - y)$ and $W$ is a white noise over $\Omega$, i.e. $(W(x))_{x \in \Omega}$ are i.i.d. $\mathcal{N}(0, 1)$ random variables.

Given an image $u \in \mathbb{R}^\Omega$, a microtexture model can be derived considering

$$m_u = \sum_{x \in \Omega} u(x)/|\Omega|, \quad \text{and} \quad U = |\Omega|^{-1/2}(u - m_u) \ast W.$$ 

(3.1)

Note that if $U$ is given by (3.1) we have for any $x, y \in \Omega$

$$\mathbb{E}[U(x)] = 0 \quad \text{and} \quad \text{Cov}[U(x), U(y)] = |\Omega|^{-1} \sum_{z \in \Omega} (\hat{u}(z) - m_u)(\hat{u}(z - (y - x)) - m_u).$$ 

(3.2)

We refer to [27] for a mathematical study of this model.

Figure 1: Examples of microtexture models. In (a) we present an original $256 \times 256$ image. In (b) and (c) we derive two microtexture models. In (b) we present a Gaussian white noise and in (c) the microtexture model given by (3.1). Note that (c) shows more local structure than (b).
3.2. Detection algorithm. In this section, \( \Omega \) is a finite square domain in \( \mathbb{Z}^2 \). We fix \( \omega \subset \Omega \). We also define \( f \), a function over \( \Omega \). We consider the Gaussian random field \( U = f * W \), where \( W \) is a Gaussian white noise over \( \Omega \). We denote by \( \Gamma_f \) the autocorrelation of \( f \), i.e. \( \Gamma_f = f * \hat{f} \) where for any \( x \in \Omega \), \( \hat{f}(x) = f(-x) \). We introduce the offset correlation function \( \Delta_f \) defined for any \( t, x \in \Omega \) by

\[
\Delta_f(t, x) = 2\Gamma_f(x) - \Gamma_f(x + t) - \Gamma_f(x - t) .
\]

The following proposition, proved in [15], gives the explicit probability distribution function of the squared \( \ell^2 \) auto-similarity.

**Proposition 2** (Squared \( \ell^2 \) auto-similarity function exact probability distribution function). Let \( \Omega = [0, M - 1] \times [0, M - 1] \) with \( M \in \mathbb{N}\setminus\{0\} \), \( \omega \subset \Omega \), \( f \in \mathbb{R}^\Omega \) and \( U = f * W \) where \( W \) is a Gaussian white noise over \( \Omega \). Then, for any \( t \in \Omega \), \( AS(U,t,\omega) \) has the same distribution as \( \sum_{k=0}^{\lfloor |\omega| \rfloor - 1} \lambda_k(t, \omega) Z_k \), with \( Z_k \) independent chi-square random variables with parameter 1 and \( \lambda_k(t, \omega) \) the eigenvalues of the covariance matrix \( C_t \) associated with function \( \Delta_f(t, \cdot) \) restricted to \( \omega \), defined in (3.3), i.e. for any \( x_1, x_2 \in \omega \), \( C_t(x_1, x_2) = \Delta_f(t, x_1 - x_2) \).

As a consequence if \( f = \delta_0 \), i.e. \( U \) is a Gaussian white noise, and \( \{x + t, x \in \omega\} \cap \omega = \emptyset \), i.e. there is no overlapping between the patch domain \( \omega \) and its shifted version, then \( AS(U,t,\omega) \) is a chi-square random variable with parameter \( |\omega| \).

In order to compute the cumulative distribution function of a quadratic form of Gaussian random variables we must deal with two issues: 1) the computation of the eigenvalues \( \lambda_k(t, \omega) \) might be time-consuming and efficient methods must be developed ; 2) the exact computation of the cumulative distribution function of a quadratic form of Gaussian random variables requires the use of heavy integrals, see [36]. In [15] a projection method is introduced in order to easily compute approximated eigenvalues, with equality when \( \omega = \Omega \). The so-called Wood F method (see [66, 3]) shows the best trade-off between accuracy and computational cost to approximate the cumulative distribution function of quadratic forms in Gaussian random variables with given weights. It is a moment method of order 3, fitting a Fisher-Snedecor distribution to the empirical one. Note that in [44] another moment method of order 3 is proposed. In what follows, we assume that we can compute the cumulative distribution function of \( AS(U,t,\omega) \) and we refer to [15] for further details.

In Algorithm 3.1 we propose an a contrario framework for spatial redundancy detection. We suppose that \( u \) and \( \omega \) are provided by the user. Using Proposition 1 and (2.6) , we say that an offset is detected if \( AP(t, \omega, AS(u,t,\omega)) \leq NFA_{\text{max}} / |\Omega| \). The value \( NFA_{\text{max}} \) is supposed to be set by the user. The background model used in the auto-similarity detection is the one given in (3.1). Therefore, Proposition 2 and the discussion that follows can be used to compute an approximation of \( AP(t, \omega, AS(u,t,\omega)) \). In Figure 2 we apply Algorithm 3.1 to a texture image.

4. Denoising.

4.1. NL-means and a contrario framework. In this section we apply the a contrario framework to the context of image denoising and propose a simple modification of the celebrated image denoising algorithm Non-Local Means (NL-means). This algorithm was introduced in the seminal paper of Buades et al. [5] and was inspired by the work of Efros and
Algorithm 3.1 Auto-similarity detection

1: function AUTOSIM-DETECTION(u, ω, NFA_{max})
2:     for t ∈ Ω do
3:         val ← AS(u, t, ω)
4:         P_{map}(t) ← AP(t, ω, val) \text{ (AP(t, ω, val) approximation detailed in Section 3.2)}
5:         D_{map}(t) ← 1_{P_{map}(t) \leq NFA_{max}/|Ω|}
6:     end for
7:     return the images P_{map}, D_{map}
8: end function

Figure 2: Outputs of Algorithm 3.1. In (a) we present an original 256 × 256 image. In (b) we present the associated microtexture model given by (3.1). In (c) the green patch is the input patch, i.e. P_ω(u). In this experiment NFA_{max} is set to 1. In (d), respectively (e), we present the output P_{map}, respectively D_{map}, of Algorithm 3.1. In (c) we show in red the patches corresponding to the identified offsets in P_{map}.

Leung in texture synthesis [25]. It was also independently introduced in [2]. This algorithm relies on the simple idea that denoising operations can be conducted in the lifted patch space. In this space the usual Euclidean distance acts as a good similarity detector and we can obtain a denoised patch by averaging all the patches with weights that depend on this Euclidean distance. Usually the weight function is set to have exponential decay, but it was suggested in [30, 57, 23] to use compactly supported weight functions in order to avoid the loss of isolated details. Since its introduction, many algorithms derived from NL-means have been proposed in order to embed the algorithm in general statistical frameworks [22, 40] or to take into account the underlying geometry of the patch space [34]. Among the state-of-the-art denoising algorithms, see [41] for a review, we consider Block-Matching and 3D Filtering (BM3D) [12] to compare our algorithm with.

There exist several works combining a contrario models and denoising tasks. Coupier et al. in [9] propose to combine morphological filters and a testing hypothesis framework to remove impulse noise. In [18] Delon and Desolneux compare different statistical frameworks to perform denoising with Gaussian noise or impulse noise. The a contrario model was also successfully used to deal with speckle noise [26] and quasi-periodic noise [61], and rely on the thresholding of wavelet or Fourier coefficients. In [37], Kervrann and Boulanger derive
approximated probabilistic thresholds using $\chi^2$ probability distribution functions. In [67] the authors propose a testing framework in order to estimate thresholds. The expressions they derive also relies on an approximation of the probability distribution of the squared Euclidean norm between two patches in Gaussian white noise.

Following a standard extension procedure of the NL-means algorithm we consider a threshold version of it, see Algorithm 4.1. In what follows we fix a “clean”, or original, image $u_0$ defined over $\Omega$, a finite rectangular domain of $\mathbb{Z}^2$, a noisy image $u = u_0 + \sigma w$, with $w$ a realization of a standard Gaussian random field $W$ and $\sigma > 0$ the standard deviation of the noise. In all of our experiments we suppose that $\sigma$ is known. Note that there exist several algorithms to estimate $\sigma$ from real images, see [54] for instance. Our goal is to retrieve $u_0$ based on the information in $u$. We consider the lifted version of $u$ in a patch space. Let $\omega_0$ be a centered $8 \times 8$ patch domain. For a patch window $\omega = x + \omega_0$ the patch search window $T$ will be defined by

$$T = \{ t \in \mathbb{Z}^2, \ t + \omega \subset \Omega, \ |t|_\infty \leq c \} ,$$

with $c \in \mathbb{N}$. $|T|$ denotes the cardinality of $T$. There exists a large literature concerning the setting of $c$ and $\omega_0$, see [23]. Note that the locality of the patch window was assessed to be a crucial feature of NL-means [31]. Suppose we have a collection of denoised patches $\hat{p}(u, \omega)$ for all patch domains $\omega$, we obtain a pixel at position $x$ in the denoised image $\hat{u}$ using the following average, see [6],

$$\hat{u}(x) = \frac{1}{|\{ t \in \Omega, \ s.t \ x \in t + \omega \subset \Omega \}|^{-1}} \sum_{t \in \Omega, \ s.t \ x \in t + \omega \subset \Omega} \hat{p}(u, t + \omega)(x) .$$

We now introduce our modification of NL-means. We suppose that we are provided a threshold function $a$. The choice of such a function is discussed in Proposition 3.

**Algorithm 4.1 NL-means threshold**

1: function NL-MEANS-THRESHOLD($u$, $\sigma$, $\omega_0$, $c$, $a$)
2: for $x \in \mathbb{Z}^2$, $x + \omega_0 \subset \Omega$ do
3: $\omega \leftarrow x + \omega_0$
4: $T \leftarrow$ defined by (4.1)
5: $N_{\omega}(u) \leftarrow 0$
6: $\hat{p}(u, \omega) \leftarrow 0$
7: for $t \in T$ do
8: if $A_S(u, t, \omega) \leq \sigma^2 a(t)$ then $\Rightarrow$ always true for $t = 0$
9: $\hat{p}(u, \omega) \leftarrow \frac{N_{\omega}(u)}{N_{\omega}(u) + 1} \hat{p}(u, \omega) + \frac{1}{N_{\omega}(u) + 1} P_{t + \omega}(u) \Rightarrow P_{\omega}(u)$ is given in Definition 1
10: $N_{\omega}(u) \leftarrow N_{\omega}(u) + 1$
11: end if
12: end for
13: end for
14: $\hat{u} \leftarrow$ defined by (4.2)
15: return $\hat{p}(u, \cdot)$, $\hat{u}$
16: end function
Note here that the output denoised version of the patch $\hat{p}(u, \omega)$ verifies the following equation

$$
\hat{p}(u, \omega) = \sum_{t \in T} \lambda_t P_{t+\omega}(u), \quad \lambda_t = \frac{\mathbb{1}_{\text{AS}(u,t,\omega) \leq a(t)}}{\sum_{s \in T} \mathbb{1}_{\text{AS}(u,s,\omega) \leq a(s)}}.
$$

In the original NL-means method, we have

$$
(4.3) \quad \lambda_t = \exp \left( -\frac{\text{AS}(u,t,\omega)}{h^2} \right) \frac{h^2}{\sum_{s \in T} \exp \left( -\frac{\text{AS}(u,s,\omega)}{h^2} \right)}. 
$$

Setting $h$ is not trivial and depends on many parameters (patch size, search window size, content of the original image). As in Algorithm 4.1, we denote $N_\omega(u) = \sum_{t \in T} \mathbb{1}_{\text{AS}(u,t,\omega) \leq a(t)}$. The following proposition, similar to Proposition 1, gives a method for setting $a$. We say that an offset $t$ is a false alarm in a Gaussian white noise if the associated patch is not used in the denoising algorithm. In Proposition 3 we choose $a$ in order to control the number of false alarms with high probability.

**Proposition 3.** Let $\text{NFA}_{\max} \in [0, |T|]$, $T$ given in (4.1) and let $a \in \mathbb{R}^\Omega$ be defined for any $t \in \Omega$ by

$$
a(t) = \mathbb{A}^{-1}(t, \omega, 1 - \text{NFA}_{\max}/|T|),
$$

with background model being a Gaussian white noise $W$, i.e. $f = \delta_0$ in Definition 4. Let $T$ be defined in (4.1) and $N_\omega(W) \in \{0, \ldots, T\}$ the random number of selected patches used to denoise the patch $P_\omega(W)$, see Algorithm 4.1. Then for any $n \in \mathbb{N}\backslash\{0\}$ it holds that

$$
\mathbb{P}_0[|T| - N_\omega(W) \geq n] \leq \frac{\text{NFA}_{\max}}{n}.
$$

**Proof.** Using the Markov inequality, we have

$$
\mathbb{P}_0[|T| - N_\omega(W) \geq n] \leq \frac{|T| - \sum_{t \in T} \mathbb{E}\left[\mathbb{1}_{\text{AS}(W,t,\omega) \leq a(t)}\right]}{n} \leq \frac{\text{NFA}_{\max}}{n}. \quad \blacksquare
$$

In the case the null hypothesis $\mathbb{P}_0$ is given by a standard Gaussian random field, which is a special case of the Gaussian random field models introduced in Section 3. In the next proposition, using the *a contrario* framework, we obtain probabilistic guarantees on the distance between the reconstructed patch $\hat{p}(u, \omega)$ and the true patch $P_\omega(u_0)$.

**Proposition 4.** Let $U = u_0 + \sigma W$, where $W$ is a standard Gaussian white noise over $\Omega$, $u_0 \in \mathbb{R}^\Omega$ and $\sigma > 0$. Let $x \in \Omega$ and $\omega = x + \omega_0$ be a fixed patch and let $\text{NFA}_{\max} \in [0, |T|]$. We introduce the random set $\hat{T} = \{t \in T, \text{AS}(U,t,\omega) \leq \sigma^2 a(t)\}$ (the selected offsets) with $a(t) = \mathbb{A}^{-1}(t, \omega, 1 - \text{NFA}_{\max}/|T|)$ as in Proposition 3 and $T$ defined in (4.1). Let $a_T = \max_{t \in \hat{T}} a(t)$. Then for any $a_W > 0$, setting $\varepsilon_W = 1 - \mathbb{P}\left[\|P_\omega(W)\|_2^2 \leq a_W | \hat{T}\right]$, we have

$$
(4.4) \quad \mathbb{P}\left[\|\hat{p}(U, \omega) - P_\omega(u_0)\|_2 \leq \sigma(a_T^{1/2} + a_W^{1/2}) | \hat{T}\right] \geq 1 - \varepsilon_W.
$$
Proof. We have for any \( t \in \hat{T} \)

\[
\|P_{t+\omega}(U) - P_\omega(u_0)\|_2 \leq \|P_{t+\omega}(U) - P_\omega(U) + P_\omega(U) - P_\omega(u_0)\|_2 \\
\leq \|P_{t+\omega}(U) - P_\omega(U)\|_2 + \|P_\omega(U) - P_\omega(u_0)\|_2 \\
\leq \sigma a_{T/2}^1 + \sigma \|P_\omega(W)\|_2.
\]

This gives the following event inclusion for any \( t \in \hat{T} \),

\[
\left\{ \|P_\omega(W)\|_2 \leq a_{W}^{1/2} \right\} \subset \left\{ \|P_{t+\omega}(U) - P_\omega(u_0)\|_2 \leq \sigma (a_{T/2}^1 + a_{W}^{1/2}) \right\}.
\]

We also have that by definition of \( \varepsilon_W \)

\[
P \left[ \|p(U, \omega) - P_\omega(u_0)\|_2 \leq \sigma (a_{T/2}^1 + a_{W}^{1/2}) \mid \hat{T} \right] \\
\geq P \left[ \bigcap_{t \in \hat{T}} \{ \|P_{t+\omega}(U) - P_\omega(u_0)\|_2^2 \leq \sigma^2 (a_{T/2}^1 + a_{W}^{1/2})^2 \} \mid \hat{T} \right] \\
\geq P \left[ \|P_\omega(W)\|_2^2 \leq a_W \mid \hat{T} \right] \geq 1 - \varepsilon_W.
\]

In our applications we use Algorithm 4.1 with \( a(t) = \text{AP}^{-1}(t, \omega, 1 - \text{NFA}_{\text{max}}/|T|) \). Therefore we need to compute \( a(t) = \text{AP}^{-1}(t, \omega, 1 - \text{NFA}_{\text{max}}/|T|) \) with a Gaussian white noise background model. We recall that in Section 3.2, using Proposition 2, we give a method to compute this quantity in general Gaussian settings. In the case of a Gaussian white noise, the next proposition shows that the eigenvalues can be computed without approximation.

Proposition 5. Let \( t = (t_x, t_y) \in \mathbb{Z}^2 \setminus \{0\} \), \( C_t \) as in Proposition 2 with \( f = \delta_0 \) and \( \omega = [0, p - 1]^2 \), with \( p \in \mathbb{N} \). We have, expressing \( C_t \) in the basis corresponding to the raster scan order on the x-axis

\[
C_t = \begin{pmatrix} B_0 & B_1 & \ldots & B_{p-1} \\ B_1^T & B_0 & \ddots & \vdots \\ \vdots & \ddots & B_0 & B_1 \\ B_{p-1}^T & \ldots & B_1^T & B_0 \end{pmatrix} + 2I_d, \quad \left\{ \begin{array}{ll} B_\ell = D_{|t_y|} \in \mathcal{M}_p(\mathbb{R}) & \text{if } \ell = |t_x| \\ B_\ell = 0 & \text{otherwise} \end{array} \right.
\]

where \( D_j \) is a zero matrix with ones on the \( j \)-th diagonal. The eigenvalues of \( C_t \) are given by \( \lambda_{m,k} = 4 \sin^2 \left( \frac{\pi m}{2p} \right) \) with multiplicity \( r_{m,k} \) where \( m \in \llbracket 2, q + 1 \rrbracket, k \in \llbracket 1, m - 1 \rrbracket \) and \( q = \left\lfloor \frac{p}{|t_x|} \right\rfloor \).

For any \( m \in \llbracket 2, q + 1 \rrbracket, k \in \llbracket 1, m - 1 \rrbracket \) it holds

(a) for any \( k' \in \llbracket 1, m - 1 \rrbracket \), \( r_{m,k} = r_{m,k'} \);
(b) \( r_{m,k} = 2|t_x||t_y| \) if \( 2 \leq m < q \);
(c) \( r_{m,k} = r_{x} r_{y} \) if \( m = q + 1 \);
(d) \( \sum_{m=2}^{q+1} \sum_{k=1}^{m-1} r_{m,k} = p^2 \),

with \( r_x = \left( \left\lfloor \frac{p}{|t_x|} \right\rfloor - q \right) |t_x| + |t_x| - p_x \), where \( p_x = |t_x| \left\lfloor \frac{p}{|t_x|} \right\rfloor - p \). We define \( r_y \) in the same manner. A similar proposition holds if \( t_y \neq 0 \).
Figure 3: **Thresholds dependency in $N_{FA}^{\max}$.** In (a) we display $a(t) = AP^{-1}(t, \omega, 1 - N_{FA}^{\max}/|T|)$ as a function of $N_{FA}^{\max}$. The patch size is fixed to $8 \times 8$ and the offsets $t$ satisfy $\|t\|_\infty \leq 10$, hence $|T| = 21^2 = 441$. The red dashed line is given by $\max_{t \in T} a(t)$ and the green dashed line by $\min_{t \in T} a(t)$. The blue line represents the value obtained considering the simplifying assumption that patch domains do not overlap, see Proposition 2 and the remark that follows. The maximal increase between the maximum of $a(t)$ and the minimum of $a(t)$ is of 13.0%. In (b) we display the mapping $t \mapsto a(t)$ for $N_{FA}^{\max} = 0.5$, the central pixel corresponds to $t = 0$. Note that $a(t)$ decreases as $\|t\|$ increases and is constant when, $\|t\|_\infty \geq 8$.

**Proof.** The proof is postponed to Appendix A.

This property allows us to compute exactly the eigenvalues appearing in Proposition 2. In Figure 3 we illustrate that $a(t)$ for fixed patch size ($8 \times 8$) and patch search window ($21 \times 21$). Thus in our implementation we suppose that $a(t) \approx \bar{a}$ and set its value to the mean of $a(t)$ over $t \in T$.

### 4.2. Some experimental results.

In the following paragraph we present and comment some results of our threshold NL-means algorithm, see Algorithm 4.1. We recall that we use $a(t) = \sum_{t \in T} AP^{-1}(t, \omega, 1 - N_{FA}^{\max}/|T|)/|T|$. In Figure 4 we present a first comparison with the NL-means algorithm. Perceptual results as well as Peak Signal to Noise Ratio (PSNR) measurements $^1$ are commented. We also present the running time of the original NL-means algorithm and ours. The experiments were conducted with the following computer specifications: 16G RAM, 4 Intel Core i7-7500U CPU (2.70GHz). Results on other images than Barbara are displayed in Figure 5.

If the threshold $a(t)$ is high, i.e. $N_{FA}^{\max} \ll |T|$ then almost no patch is rejected, which means that almost all patches are used in the denoising process. In consequence, the output denoised image is very smooth. This smoothness is a correct guess for constant patches. However, this proposition does not hold when the region contains details. Indeed, in this

$^1$PSNR($u, v$) = $10 \log_{10} \left( \frac{\text{max}_{u} u^2}{\text{max}_{u-v} u^2} \right)$. 

Figure 4: **Visual results.** In (a) we present an original image (Barbara) scaled between 0 and 255. In (b) we add Gaussian white noise with $\sigma = 10$. We recall that the patch domain is fixed to $\omega_0$ being a $8 \times 8$ square. In (c) we present the denoised results with NL-means threshold, Algorithm 4.1, where $\text{NFA}_{\text{max}} = 4.41$, which corresponds to 1% of rejected patches in the search window of a Gaussian white noise. In (d) we present the results obtained with the traditional NL-means algorithm with $h = 0.13\sigma|\omega|$ (optimal $h$ for this noise level and this image with regard to the PSNR measure). The results are the same on the texture area for (c) and (d). The perceptual results on the zoomed region are satisfying, even though some regions are too smooth compared to the original image (a). In (c) and (d), $\delta_t$ is the running time of the algorithm. We can observe that our algorithm is slightly slower than NL-means.

case details are lost due to the averaging process. By setting a conservative threshold, e.g. $\text{NFA}_{\text{max}} / |T| \approx 0.1$, for example, we reject all the patches for which the structure does not strongly match the one of the input patch, see Figure 6. This conservative property of the algorithm ensures that we can control the loss of information in the denoised image, see Proposition 4. However, if no patch, other than the input patch itself, is detected as similar we highly overfit the original noise. Many algorithms such as BM3D, see [12], solve this problem by treating this case as an exception, applying a specific denoising method in this situation. We show the differences between our version of NL-means and BM3D in Figure 7.

In Figure 8, we show that Algorithm 4.1 performs better than the original NL-means algorithm. By setting $\text{NFA}_{\text{max}} / |T| = 0.01$ we obtain that the PSNR of the denoised image is better than the one of NL-means for nearly every value of $h$.

Let us emphasize that our goal is not to provide a new state-of-the-art denoising algorithm. Indeed we never obtain better denoising results than the BM3D algorithm. However, our algorithm slightly improves the original NL-means algorithm. It shows that statistical testing can be efficiently used to measure the similarity between patches and therefore provides a robust way to perform the weighted average in this algorithm.

5. Periodicity analysis.

5.1. Existing algorithms. In the following sections we use our patch similarity detection algorithm, see Algorithm 3.1, to analyze images exhibiting periodicity features. Let $\Omega \subset \mathbb{Z}^2$ be a finite domain and $\omega \subset \Omega$ a finite patch domain.

Periodicity detection is a long-standing problem in texture analysis [71]. First algorithms
Figure 5: **NL-means comparison.** In this figure we compare Algorithm 4.1 with the traditional NL-means algorithm. Here $\omega_0$ is fixed to be a $8 \times 8$ square. The first column contains clean images, the second column represents the same images corrupted by a Gaussian noise with $\sigma = 20$. The third column is the output of Algorithm 4.1 with $NFA_{\max}$ fixed to 4.41 and the last column is the output of the NL-means algorithm for the optimal value of $h$ (with regards to the PSNR), see (4.3). Perceptual results and PSNR are comparable, even though our algorithm yields slightly better PSNR values. We also present the running times $\delta t$ of both algorithm. Our algorithm is slower than NL-means as it computes the threshold before running the NL-means algorithm.
Figure 6: **Number of detections.** In this figure we present, for each denoised pixel, the number of detected offsets used to compute the denoised patch, i.e. the cardinality of $\hat{T}$, see Proposition 4. A white pixel means that the number of detected offsets is maximal and a black pixel means that the number of detected offsets is $1$, i.e. the patch is not denoised. As $\text{NFA}_{\text{max}}$ decreases the number of detected offsets increases. Note that $|\hat{T}|$ is maximal, i.e. equals to $21^2 = 441$, for constant regions. For $\text{NFA}_{\text{max}} = 0.1$, pixels located in textured neighborhoods use in average 20 to 40 patches to perform denoising.

used the quantization of images, relying on co-occurrence matrices and statistical tools like $\chi^2$ tests or $\kappa$ tests. Global methods extract peaks in the frequency domain (Fourier spectrum) \[49\] or in the spatial domain (autocorrelation). In \[32\] the notion of inertia is introduced. It is defined for any $t \in \Omega$ by $I_t = \sum_{(i,j) \in [0,N_g]^2} (i - j)^2 \mathbb{1}_{u(x)=i, 
abla u(x+t)=j}$, where $u$ is a quantized image on $N_g + 1$ gray levels. In \[8\], the authors show that the local minima of the inertia measurement can be used to assess periodicity. Similarly, we introduce the $\omega$-inertia for any $t \in \Omega$ by $I_{\omega,t} = \sum_{(i,j) \in [0,N_g]^2} (i - j)^2 \mathbb{1}_{u(x)=i, \nabla u(x+t)=j}$. The following proposition extends to a local framework results from \[52\].

**Proposition 6.** Let $u \in \mathbb{R}^\Omega$. Suppose that $u$ is quantized, i.e. there exists $N_g \in \mathbb{N}$ such that for any $x \in \Omega$, $u(x) \in [0, N_g]$. We have $I_{\omega,t} = \mathcal{AS}(u, t, \omega)$.

**Proof.** For any $t \in \Omega$ we have

$$I_{\omega,t} = \sum_{(i,j) \in [0,N_g]^2} (i - j)^2 \mathbb{1}_{u(x)=i, \nabla u(x+t)=j} = \sum_{x \in \omega, (i,j) \in [0,N_g]^2} (i - j)^2 \mathbb{1}_{u(x)=i, \nabla u(x+t)=j}$$

$$= \sum_{x \in \omega} (\hat{u}(x) - \hat{u}(x+t))^2 = \mathcal{AS}(u, t, \omega).$$

If $\omega = \Omega$ then the $\omega$-inertia statistics is exactly the inertia introduced in \[32\] and the result is due to \[52\].

**5.2. Algorithm and properties.** Lattice detection is closely related to periodicity analysis, since identifying a lattice is similar to extracting periodic or pseudo-periodic structures up to deformations and approximations. A state-of-the-art algorithm proposed in \[53\] uses a
Figure 7: Comparison with BM3D. We compare Algorithm 4.1 to BM3D [12]. The original image (Barbara) is presented in (a). We consider a noisy version of the input image with $\sigma = 20$. In (b) we present the output of BM3D, with default parameters, see [39]. The result in (c) corresponds to the output of Algorithm 4.1 with $\text{NFA}_{\text{max}} / |T| = 0.01$. The output (c) is too blurry compared to (b). In order to correct this behavior we set $\text{NFA}_{\text{max}} / |T| = 0.1$ in (d), i.e. increase the global threshold and some improvements are noticeable. However the image remains blurry and artifacts due to the overfitting of the noise appear, this is known as the rare patch effect in [22]. For instance, some patches in the scarf are not denoised anymore.

Figure 8: PSNR study. In this figure we present the evolution of the PSNR for different values of the parameter $h$ of the original NL-means method, see (4.3), in blue, computed on the Barbara image. The $x$-axis represents $h / \sigma |\omega|$. The orange dashed line is the PSNR obtained for the threshold NL-means algorithm (Algorithm 4.1) with $\text{NFA}_{\text{max}} / |T| = 0.01$. Except for low levels of noise the proposed method gives better PSNR values than the original implementation of NL-means algorithm for every choice of $h$.

recursive framework which consists in 1) a lattice model proposal based on detectors such as Kanade-Lucas-Tomasi (KLT) feature trackers [48], 2) spatial tracking using inference in a probabilistic graphical model, 3) spatial warping correcting the lattice deformations in the original image. In this section we propose a new algorithm for lattice detection. The lattice
proposal step 1) is replaced by an Euclidean auto-similarity matching detection (see Section 3.2 and Algorithm 3.1) where the patch domain \( \omega \) is fixed. Using these detections we build a graph with a few nodes (usually approximately 20 nodes for a 256 \( \times \) 256 image). We use the same notation for the detection mapping \( t \mapsto 1_{\mathcal{A}S((u,t,\omega)\leq a(t))} \) i.e. the \( D_{\text{map}} \) output of Algorithm 3.1, which is a binary function over the offsets, and the set of detected offsets. We recall that two pixel coordinates \( x \) and \( y \) are said to be 8-connected if \( x = y + (\delta_x, \delta_y) \) with \( \delta_x, \delta_y \in \{-1,0,1\} \). The graph \( \mathcal{G} = (V,E) \) is then built as follows:

- **Vertices:** for each 8-connected component, \( \mathcal{C}_k \) in \( D_{\text{map}} \) we note \( v_k \) one position for which the minimum of \( \mathcal{A}S(u,t,\omega) \) over \( \mathcal{C}_k \) is achieved. The set of vertices \( V \) is defined as \( V = \{v_{k}\}_{k \in [1,N_{\mathcal{C}}} \) where \( N_{\mathcal{C}} \) is the number of 8-connected components in \( D_{\text{map}} \);
- **Edges:** each vertex is linked with its four nearest neighbors in the sense of the Euclidean distance, defining four unoriented edges.

Refering to the three steps of [53] we present our model to replace step 2) (i.e. the inference in a probabilistic graphical model) and introduce the approximated lattice hypothesis defined on a graph.

**Definition 5 (Approximated lattice hypothesis).** Let \( \mathcal{G} = (V,E) \) be a random graph with \( V \subset \mathbb{R}^2 \). We say that \( \mathcal{G} \) follows the approximated lattice hypothesis if there exists a basis \( B = (b_1, b_2) \) of \( \mathbb{R}^2 \) and, for each edge \( e \in E \), a couple of integers \((m_e, n_e) \in \mathbb{Z}^2 \) such that \( e \) has the same distribution as \( m_e b_1 + n_e b_2 + \sigma \mathcal{Z}_e \), with \((\mathcal{Z}_e)_{e \in E} \) independent standard Gaussian random variables in \( \mathbb{R}^2 \) and \( \sigma > 0 \). We denote by \( \mathbf{M} \) the vector of all coefficients \((m_e, n_e)_{e \in E} \in \mathbb{Z}^{|E|} \).

Our goal is to perform inference in the statistical model defined by the following log-posterior

\[
\mathcal{L}(\mathbf{B}, \mathbf{M}, \sigma^2|E) = -\frac{1}{2\sigma^2} \sum_{e \in E} \|m_e b_1 + n_e b_2 - e\|^2 + r(\mathbf{B}, \mathbf{M}),
\]

where \( r(\mathbf{B}, \mathbf{M}) = \delta_B\|\mathbf{B}\|_2^2 + \delta_M\|\mathbf{M}\|_2^2 \) with \( \delta_B, \delta_M \geq 0 \). A discussion on the dependence of the model on the hyperparameters \((\delta_B, \delta_M)\) is conducted in Figure 9. Finding the MLE of this full log-posterior is a non-convex, integer problem. However performing the minimization alternatively on \( \mathbf{B} \) and \( \mathbf{M} \) is easier since at each step we only have a quadratic function to minimize. Minimizing a positive-definite quadratic function over \( \mathbb{Z}^2 \) is equivalent to finding the vector of minimum norm in a lattice. This last formulation is known as the Shortest Vector Problem (SVP) which is a challenging problem [51] (though it is not known if it is a NP-hard problem). We replace this minimization procedure over a lattice by a minimization over \( \mathbb{R}^2 \) followed by a rounding of this relaxed solution.

For any \( \sigma > 0 \) we denote by \( \mathcal{L}_n(\sigma) = \mathcal{L}(\mathbf{B}_n, \mathbf{M}_n, \sigma^2|E) \), with \( n \in \mathbb{N} \), the log-posterior sequence.

**Proposition 7 (Alternate minimization update rule).** In Algorithm 5.1, we get for any \( n \in \mathbb{N} \)

\[
\mathbf{M}_n = (\Lambda_{B_n} \otimes \text{Id}_{|E|})^{-1} E_{B_n} \in \mathbb{R}^{2|E|}, \quad B_{n+1} = (\Lambda_{M_{n+1}} \otimes \text{Id}_2)^{-1} E_{M_{n+1}} \in \mathbb{R}^4,
\]

with \( \otimes \) the tensor product between matrices and

(a) \( \Lambda_B = \left( \frac{\|b_1\|^2 + \delta_B}{\langle b_1, b_1 \rangle}, \frac{\langle b_1, b_2 \rangle}{\|b_2\|^2 + \delta_B} \right) \), \( \Lambda_M = \left( \frac{\|M_1\|^2 + \delta_M}{\langle M_1, M_1 \rangle}, \frac{\langle M_1, M_2 \rangle}{\|M_2\|^2 + \delta_M} \right) \);
Algorithm 5.1 Lattice detection – Alternate minimization

1: function ALTERNATE-MINIMIZATION(\(E, \delta_B, \delta_M, N_{it}\))
2: \(M_0 \leftarrow 0\)
3: \(B_0 \leftarrow \) initialization procedure \(\triangleright\) initialization discussed in the end of Section 5.2
4: for \(n \leftarrow 0\) to \(N_{it} - 1\) do
5: \(M \leftarrow \arg\min_{M \in \mathbb{R}^2} q(B_n, M|E)\) \(\triangleright\) expression given in Proposition 7
6: if \(q(E|B_n, [M]) < q(E|B_n, M_n)\) then
7: \(M_{n+1} \leftarrow [M]\)
8: else
9: \(M_{n+1} \leftarrow M_n\)
10: end if
11: \(B_{n+1} \leftarrow \arg\min_{B \in \mathbb{R}^4} q(B, M_{n+1}|E)\) \(\triangleright\) expression given in Proposition 7
12: end for
13: \(\sigma^2_{N_{it}} \leftarrow \arg\min_{\sigma^2 \in \mathbb{R}_+} \mathcal{L}(B_{N_{it}}, M_{N_{it}}, \sigma^2|E)\)
14: return \(B_{N_{it}}, M_{N_{it}}, \sigma^2_{N_{it}}\)
15: end function

\(b)\) \(E_B = \left(\left(\langle \mathbf{e}, b_1 \rangle \right)_{\mathbf{e} \in E} \right), \quad E_M = \left(\frac{m_{\mathbf{e}} \mathbf{e}}{\sum_{\mathbf{e} \in E} n_{\mathbf{e}} \mathbf{e}}\right)\).

Proof. The proof is postponed to Appendix B.

Note that if \(B\) is orthogonal, i.e. \(\langle b_1, b_2 \rangle = 0\) then \(\Lambda_B\) is diagonal and the proposed method is the exact solution to the minimization problem over \(\mathbb{Z}^2\).

Theorem 1 (Convergence in finite time). For any \(\sigma > 0\), \((\mathcal{L}_n(\sigma))_{n \in \mathbb{N}}\) is a non-decreasing sequence. In addition, \((B_n)_{n \in \mathbb{N}}\) and \((M_n)_{n \in \mathbb{N}}\) converge in a finite number of iterations.

Proof. \((\mathcal{L}_n(\sigma))_{n \in \mathbb{N}}\) is non-decreasing since for any \(n \in \mathbb{N}\), \(\mathcal{L}_n(\sigma) \leq \mathcal{L}(B_n, M_{n+1}, \sigma^2|E) \leq \mathcal{L}_{n+1}(\sigma)\). Let us show that the sequences \((M_n)_{n \in \mathbb{N}}\) and \((B_n)_{n \in \mathbb{N}}\) are bounded. Because \((\mathcal{L}_n(\sigma))_{n \in \mathbb{N}}\) is non-decreasing, the sequence \((q(B_n, M_n|E))_{n \in \mathbb{N}}\) is non-increasing. We obtain that

\[
\delta_M\|M_n\|^2 \leq q(B_0, M_0|E), \quad \delta_B\|B_n\|^2 \leq q(B_0, M_0|E).
\]

The sequence \((M_n)_{n \in \mathbb{N}}\) is bounded thus we can extract a converging subsequence. Since \((M_n)_{n \in \mathbb{N}}\) takes value in \(\mathbb{Z}^2|E|\), this subsequence is stationary with value \(M\). Let \(n_0 \in \mathbb{N}\) be the first time we hit value \(M\). Let \(n \in \mathbb{N}\), with \(n \geq n_0 + 1\), there exists \(n_1 \in \mathbb{N}\), with \(n_1 \geq n\) such that \(M_{n_1} = M_{n_0}\) thus

\[
\mathcal{L}_{n_0}(\sigma) \leq \mathcal{L}_{n_0+1}(\sigma) \leq \mathcal{L}_n(\sigma) \leq \mathcal{L}(B_{n-1}, M_{n_1}, \sigma^2|E) \leq \mathcal{L}(B_{n-1}, M_{n_0}, \sigma^2|E) \leq \mathcal{L}_{n_0}(\sigma).
\]

Hence for every \(n \geq n_0 + 1\), \(\mathcal{L}_n(\sigma) = \mathcal{L}(B_n, M_n, \sigma^2|E) = \hat{\mathcal{L}}(\sigma)\). Suppose there exists \(n \geq n_0 + 1\) such that \(M_n \neq M_{n+1}\) this means that \(\mathcal{L}(B_n, M_{n+1}, \sigma^2|E) > \mathcal{L}_n(\sigma)\) (because
Figure 9: In this experiment we assess the importance of the hyperparameters. We consider Algorithm 5.1 with input graph a detection map, output of Algorithm 3.1. The initialization in the three cases is the canonical basis \((0,1),(1,0)\). In (a), since the initial basis vectors are a local minimum to the optimization problem, the algorithm converges after one iteration. However, this is not perceptually satisfying. Setting \(\delta_M = 5\) and \(\delta_B = 10^{-1}\) in (b) the true observed lattice is a sub-lattice of the output lattice of Algorithm 5.1. Increasing \(\delta_M\) up to 9, in (c) we obtain a perceptually correct lattice. For \(\delta_M\) larger than 10, the basis vectors go to 0. Only the regularizing term is minimized by the optimization procedure and the data-attachment term is not taken into account. Experimentally we found that the choice of \(\delta_M\) is more flexible and that \(\delta_M = 10\) gives satisfying perceptual results if the initialization heuristics proposed in Section 5.2 is chosen.

of lines 6 and 7 of Algorithm 5.1) which is absurd. Thus \((M_n)_{n\in\mathbb{N}}\) is stationary and so is \((B_n)_{n\in\mathbb{N}}\).

In Algorithm 5.1 \(M_0\) is initialized with zero and \(B_0\) is defined as an orthonormal (up to a dilatation factor) direct basis where the first vector is given by an edge with median norm in \(E\).

5.3. Experimental results. Combining the results of Section 5.2 and Section 3.2 we obtain an algorithm to extract lattices in images, see Figure 10. In what follows we perform lattice detection using Algorithm 3.1 in order to extract auto-similarity given a patch in an original image \(u\), which implies that the patch domain \(\omega\) is set by the user. Recall that in Algorithm 3.1, the eigenvalues of the covariance matrix in Proposition 2 are approximated, and that the cumulative distribution function of the quadratic form in Gaussian random variables is computed via the Wood F method [66]. Lattice detection is performed using Algorithm 5.1 with parameters \(\delta_M = 10\) and \(\delta_B = 10^{-2}\).

5.3.1. Escher paving. In this section we study art images, Escher pavings, with strongly periodic structure. We investigate the following parameters of our lattice detection algorithm:

(a) background microtexture model \(\mathcal{F}_0\),
(b) NFA\(_{\text{max}}\) parameter in Algorithm 3.1,
(c) patch domain \(\omega\).
Figure 10: Lattice proposal algorithm. Lattice detection and extraction in images require a patch from the user and compute a binary image containing all the offsets with correct similarity as well as a lattice matching the underlying graph. The patch auto-similarity detection step was presented in Section 3.2. The lattice detection step was presented in Section 5.2. The first image is the input, the second one is the output of the detection algorithm. In the last step we show the original image with red squares placed on the computed lattice. Behind this image, the unoriented edges of the graph are shown in red.

Microtexture model. We confirm that the choice of the microtexture model will influence the detected geometrical structures. The more structured is the background noise model the less we obtain detections. This situation is considered in Figure 11.

NFA_{\text{max}} parameter. Using a more adapted microtexture model as background model we gain robustness compared to other less structured models such as a Gaussian white noise. However, NFA_{\text{max}} must be set carefully otherwise two situations can occur:

(a) if NFA_{\text{max}} is too high, too many detections can be obtained (true perceptual detections are not differentiated from false positives);

(b) if NFA_{\text{max}} is too low, we fail to identify important perceptual structures in the image.

We observe that a general good practice is to set NFA_{\text{max}} equal to 10, see Figure 12. However, if the input patch is corrupted one may increase this parameter up to 10 \(^2\) or 10 \(^3\), see Figure 17 and Figure 18.

Patch position. Patch position and size are crucial in our detection model, since we rely on local properties of the image. As shown in Figure 13 these parameters should be carefully selected by the user. However, for particular applications such as lattice extraction for crystallographic purposes, there exist procedures to extract primitive cells [50].

5.3.2. Crystallography images. Defect localization, noise reduction, correction of crystalline structures in images are central tasks in crystallography. Usually, they require the knowledge of the geometry of a perfect underlying crystal. In our experiments we manually identify the geometry of the periodic crystal, which allows for multiple structures in one image, provided a user input of the primitive cell in a lattice. This primitive cell extraction could be automated [50]. In Figure 14, we present an example of multiple geometry extraction. Statistics like angle and period can be retrieved using the estimated basis vectors. This image contains two lattices and the locality of our measurements allows for the detection of both structures. Using windowed Fourier transform could be efficient to obtain local measurements on the periodicity of these images since the information is highly frequential. However in or-
Figure 11: **Choice of the microtexture model.** In this experiment we discuss the choice of the *a contrario* background microtexture model. In the left column we display the graph obtained after the detection step. In the middle column we superpose the proposed lattice on the original image. The original patch is drawn in green, obtained basis lattice vectors are in cyan, and red squares are placed onto the proposed lattice. In (a) and (b) the microtexture model is given by (3.1) and $\text{NFA}_{\text{max}}$ is set to 10. A sample of this model is presented in (c). Obtained results match the perceptual lattice. In (d), (e), (g) and (h) the microtexture model is a Gaussian white noise model with variance equal to the empirical variance of the original image. Sample from this Gaussian white noise is presented in (f). In (d) and (e), $\text{NFA}_{\text{max}}$ is set to 10. This leads to an excessive number of detections in the input image. In order to obtain the perceptual lattice found in (b) with a Gaussian white noise model we must set the $\text{NFA}_{\text{max}}$ parameter to $10^{-111}$. Results are presented in experiments (g), (h) and (i). Image (h) is also an example for which the median initialization for $B_0$ in Algorithm 5.1 identifies a non satisfying local minimum. This situation is corrected in (i) with random initialization for $B_0$. In (h) final log-posterior value is $-565.5$ which is inferior to the final log-posterior value in (i): $-542.1$. Thus (i) gives a better local maximum of the full log-posterior than (h).
Figure 12: **Choice of Number of False Alarms.** In this experiment we discuss the choice of the $NFA_{\text{max}}$ parameter in the *a contrario* framework in the case where the underlying microtexture model is given by (3.1). Each column corresponds to a pair of images: the returned lattice and its associated underlying graph. In (a), $NFA_{\text{max}}$ is set to 1. Detections are correct, there are not enough points to precisely retrieve the perceptual lattice. In (b), $NFA_{\text{max}}$ is set to 10. The estimated lattice is correct. In (c), $NFA_{\text{max}}$ is set to $10^3$. In this case we obtain false detections which lead to an incorrect final lattice. Note that large detection zones in the binary image (c) are due to the non-validity of the Wood F approximation for some offsets. This behavior is also present in (a) and (b) but less noticeable.

In order to obtain the same detection map as Algorithm 3.1 one must carefully set the threshold parameter, $NFA_{\text{max}}$. This situation is illustrated in Figure 15.

Finally we assess the precision of our measurements by comparing our results with a model used in crystallography, see Figure 16. We indeed retrieve one of the possible bases used to describe these lattices. However, the symmetry constraints are not present in the identified basis. To obtain another basis, one must relax the regularization parameters. A more natural way to obtain the desired primitive cell would be to introduce symmetry constraints in the graphical model formulation in (5.1).

**5.3.3. Natural images.** Identifying lattices in natural images is a more challenging task since we have to deal with image artifacts. In this section we investigate the effect on the detection of the background clutter in natural images, see Figure 17, and the effect of the
Figure 13: **Influence of patch size and patch position.** For each experiment $\text{NFA}_{\text{max}}$ is set to $10^4$, i.e. 4% of the pixels. In most cases lower $\text{NFA}_{\text{max}}$ could be used but setting a high $\text{NFA}_{\text{max}}$ ensures that we always get detections even if the patch only contains microtexture information. Each row corresponds to a lattice proposal with same patch position but different patch sizes: $10 \times 10$ for the left column, $15 \times 15$ for the middle one and $20 \times 20$ for the right one. Each image represents the superposed proposed lattice on the original image. On the bottom-right of each image we display the underlying graph as well as the binary detection. On the first row the patch contains only a white region with a few gray pixels. The influence of these pixels is visible for small patch sizes (a) but is no longer taken into account for larger patch sizes, (b) and (c). On the second row the patch contains gray microtexture which has some local structure. We identify large similarity regions and no perceptual lattice is retrieved in (d), (e) and (f). The situation is different on the third row. The $10 \times 10$ patch contains only uniform black information in (g), but the situation changes as the patch sizes grows. In (h), the patch intersects black, gray and white zones. The graph is much sparser and the lattice is close to the perceptual one. In (i), the patch size is large enough to cover large areas of the three gray levels and the perceptual lattice is identified.
Figure 14: **Lattice extraction.** In this experiment we consider a crystallographic image (an orthorombic NiZr alloy) and set $\text{NFA}_{\text{max}}$ to $10^2$. Two lattices are present in this image and they are correctly identified in (b) and (d). Note that in (a), respectively in (c), mostly points in the left, respectively right, part of the image are identified, thus yielding correct lattice identification. Points which should be identified and are discarded nonetheless correspond to regions in which we observe contrast variation. Image courtesy of Denis Gratias.

![Image](image1.png)

Figure 15: **Comparison with Fourier based methods.** Since the original image can be segmented in two highly periodic components, Fourier methods might be well-adapted to the lattice extraction task. In (a) we present a sub-image of the original alloy. We compute the autocorrelation of this sub-image and threshold it. This operation gives us a detection map, like Algorithm 3.1. In (b) the threshold is set to 90% percent of the maximum value of the autocorrelation. Too many points are identified. In (d) the threshold is set to 99% and only one point is identified. Correct lattice is identified in (c).

![Image](image2.png)

**Preprocessing.** Due to the occlusions occurring in natural images, if a lattice is superposed over a real photograph, carefully selecting structural elements might not be enough in order to retrieve the periodicity. Indeed, if we observe a repetition of the lattice pattern, the background does not necessarily contain any repetition and thus makes the detection more complicated. In order to avoid such a problem we propose to introduce a preprocessing step in our algorithm. This preprocessing step will be encoded in a linear filter $h$. Suppose $U$ is a sample from a
Figure 16: **Agreement with crystallography models.** In (a) we perform a zoom on of the lattice identified in Figure 14 and compare it to the one identified by crystallographists in (b). (a) is a zoomed rotated version of a crystalline structure similar to (b). The output lattice in (a) is the same as the one in (b). Indeed in (b) the red points, for instance, form a lattice. A possible basis for this lattice is given by the vectors of a parallelogram. Up to rotation these basis vectors match the one identified in (a). However, the parallelogram basis is a symmetric and thus is not chosen by chemists since it does not reflect the geometry of the alloy. The preferred basis is given by the symmetric rhombus (white edges in (b)). Image courtesy of Denis Gratias.

Gaussian model with function $f$ then $h \ast U$ is a sample from a Gaussian model with function $h \ast f$. Thus all the properties derived earlier remain valid with this linear operation. In Figure 17, we set $h$ to be a Laplacian operator $^2$. This operation allows us to avoid contrast problems.

*Homography.* In the previous experiments we suppose that the lattice structure was in front of the camera. In many cases this assumption is not true and there exists an homography that matches the deformed lattice in the image to a true lattice. Our algorithm makes the assumption that the lattice is viewed in a frontal way and fails otherwise. However, locally, this assumption is true and we can observe partial match of the lattices in Figure 18.

### 5.4. Texture ranking

We conclude these experiments by showing that this simple graphical model can be used to perform ranking among texture images, sorting them according to their degree of periodicity. We say that an image has high periodicity degree if a lattice structure can be well fitted to the image. We introduce a criterion for evaluating the relevance of the lattice hypothesis. Let $u$ be an image over $\Omega$, let $\omega \subset \Omega$ be a patch domain and $a$ be as

\[ \Delta(u)(x_1, x_2) = (u(x_1 + 1, x_2) + u(x_1 - 1, x_2) + u(x_1, x_2 + 1) + u(x_1, x_2 - 1) - 4u(x))/4, \]

where boundaries are handled periodically.

---

$^2$We use a discrete Laplacian operator $\Delta$ such that for any $x = (x_1, x_2)$, we get that $\Delta(u)(x_1, x_2) = (u(x_1 + 1, x_2) + u(x_1 - 1, x_2) + u(x_1, x_2 + 1) + u(x_1, x_2 - 1) - 4u(x))/4$, where boundaries are handled periodically.
Figure 17: \textit{Preprocessing and filtering.} In (a) and (c) we display the graphs obtained with Algorithm 3.1 applied on images (b) and (d). In (b) and (d) the original image is superposed with the estimated lattice (vectors in cyan and proposed patches in red). In (a) and (b), $\text{NFA}_{\text{max}}$ was set to $10^5$ which corresponds to 35\% of detection in the associated \textit{a contrario} model. Lower $\text{NFA}_{\text{max}}$ did not give enough points to conduct the lattice proposal step. We obtain a visually satisfying lattice. In (c) and (d) we apply a simple preprocessing, a Laplacian filter, to the image and set $\text{NFA}_{\text{max}}$ to 10. The detection figure is much cleaner and the estimation makes much more sense from a perceptual point of view. Note that, as in (b), the proposed lattice does not exactly match the fence periodicity. This is due to: 1) the initialization of the algorithm and the structure of the graph in the alternate minimization algorithm 2) the fact that the horizontal periodicity is broken by the black post.

Figure 18: \textit{Homography and locality.} In this experiment $\text{NFA}_{\text{max}}$ was set to $10^3$. Note that the detected graph is localized around the original patch in (a). In (b) we superpose the proposed lattice onto the original image. The lattice proposal is valid in a small neighborhood around the original patch. However it is not valid for the whole image.
in Proposition 1 with NFA\textsuperscript{max} set by the user.

**Definition 6 (Periodicity criterion).** Let \( \{ t \in \Omega, \ A\mathcal{S}(u, t, \omega) \leq a(t) \} \) be the set of detected offsets and \( N_{\mathcal{E}} \) its number of connected components as defined in Section 5.2. Let also \((\hat{B}, \hat{\mathcal{M}}, \hat{\sigma})\) be the estimated parameters using Algorithm 5.1. We define the following periodicity criterion \( c_{\text{per}} \) as

\[
c_{\text{per}}(u) = \frac{\pi \hat{\sigma}^2}{N_{\mathcal{E}} |\det(b_1, b_2)|},
\]

where \( \hat{B} = (b_1, b_2) \).

The criterion \( c_{\text{per}} \) simply computes the ratio between the error area of Algorithm 5.1, i.e. the error made when considering the approximated lattice hypothesis, see Definition 5, and the area of the parallelogram defined by the output basis vectors. If we have enough detections this quantity is supposed to be small when the approximated lattice hypothesis holds and large when it does not. Nonetheless, we introduce a dependency in the number of detections. Indeed, even if no lattice is perceived, the hypothesis in Definition 5 may still hold if the number of detected offsets is small.

In the experiment presented in Figure 19 we sort 25 texture images based on the \( c_{\text{per}} \) criterion. Images are of size 256 \( \times \) 256. Since the identified graph highly depends on the patch position and the patch size, for each image we uniformly sample 150 patch positions and set the patch size to 20 \( \times \) 20. For each set of parameters we find a lattice using Algorithm 3.1 and Algorithm 5.1 with parameters \( \text{NFA}_{\text{max}} = 1, \ \delta_M = 10, \ \delta_B = 10^{-2} \) and \( N_{\mathcal{U}} = 10 \). A statistical study of our ranking is presented in Figure 20. Note that, from a perceptual point of view, from (a) to (n) all textures are periodic except for (f), (j) and (k) which are examples for which our algorithm fails. However, from (o) to (y), no texture is periodic.

### 6. Conclusion

In this paper we introduce a statistical model, the *a contrario* framework, to analyze spatial redundancy in images. We propose a general algorithm for detecting redundancy in natural images. It relies on Gaussian random fields as background models and takes advantage of the links between the \( \ell^2 \) norm and Gaussian densities. The *a contrario* formulation provides us with a statistically sound way of thresholding distances in order to assess similarity between patches. In this rationale we replace the task of manually setting thresholds by the selection of a Number of False Alarms.

We illustrate our contribution with three examples in various domains of image processing. Introducing a simple modification of the NL-means algorithm we show that similarity detection (in this case, dissimilarity detection) in a theoretical *a contrario* framework can easily be embedded in any image denoising pipeline. For instance, the threshold we introduced could be integrated into the Non-Local Bayes algorithm \cite{40} in order to estimate mean and covariance matrices with probabilistic guarantees. The generality of our model allows for several extensions for non-Gaussian noises \cite{16} or to take into account the geometry of the patch space \cite{34, 63}.

Turning to periodicity detection we propose a novel graphical model using the output of Algorithm 3.1 in order to extract lattices from images. In this model, lattice extraction is formulated as the maximization of some log-likelihood defined on a graph. We prove the
Figure 19: Texture ranking. The $c_{\text{per}}$ criterion, defined in (5.2), is computed for each setting. We associate to each image the median of the 150 criterion values and sort the images accordingly. (a) corresponds to the lowest criterion, i.e. the most periodic image according to $c_{\text{per}}$ criterion. (y) corresponds to the largest criterion, i.e. the least periodic image according to $c_{\text{per}}$. Under each image we give the logarithm of the median $c_{\text{per}}$ values.
Figure 20: **Boxplot for $c_{\text{per}}$ values.** In this figure we present a boxplot of the $c_{\text{per}}$ values, defined in (5.2), used to rank textures images in Figure 19. We recall that we use 150 random patch positions in order to compute the $c_{\text{per}}$ values. Letters on the $x$-axis correspond to the textures in Figure 19. For each texture we present its median $c_{\text{per}}$ value. The lower, respectively upper, limit of the blue box corresponds to 25%, respectively 75% of the computed $c_{\text{per}}$ values. The dashed line corresponds to the confidence interval with level 0.07 under normality assumption. Points outside this interval are plotted in red and the graphics was clipped between $0$ and $5 \times 10^{-3}$. The size of the confidence interval grows with the median value. It must be noted that the overlapping of the blue boxes might explain some inconsistencies of our ranking. Another source of errors lie in the model which assumes that if a texture is periodic its pattern is described by a $20 \times 20$ patch. In order to perform a more robust ranking a multiscale approach should be preferred.

finite-time convergence of Algorithm 5.1. We provide image experiments illustrating the role of the hyperparameters in our study and we assess the importance of selecting adaptive Gaussian random fields as background models. We remark that the expected number of false alarms parameter is linked to the choice of the input patch and give a range of possible values for $\text{NFA}_{\text{max}}$ settings. We also illustrate its possible application in crystallography as it correctly identifies underlying lattices in alloys. This rationale could be used to identify symmetry groups (wallpaper groups) in alloys, following the work of [45]. Finally our method is tested on natural images where some of its limits such as perspective defect or sensitivity to occlusion phenomena are identified. It must be noted that our method could easily be extended to color images by considering $\mathbb{R}^3$-valued instead of real-valued images.

Our last application consists in giving a quantitative criterion for periodicity texture ranking. This criterion is based on the parameters estimated in Algorithm 5.1. Since we set our background models to be Gaussian random fields and remarking that these are good micro-texture approximations we wish to explore the possibility to embed our *a contrario* framework...
Thus the one-dimensional vector (given by the raster-scan order on the $x$-axis) of the restriction of $X_{x_0,k}$ is an eigenvector of $C$ associated with eigenvalue $4 \sin^2 \left( \frac{k\pi}{m} \right)$. Using that $f$ is in bijection with $\Omega$ we get that the number of vectors $(X_{x_0,k})$ is $|\Omega|$. We show that this family of vectors is linearly-independent. Let $\Lambda_{x_0,k} \in \mathbb{R}$ such that

$$
\sum_{x_0 \in \Omega_0} \sum_{k=1}^{J(x_0)-1} \Lambda_{x_0,k} X_{x_0,k} = 0.
$$

Since $X_{x_0,k}$ and $X_{y_0,k'}$ have different support if and only if $x_0 \neq y_0$ we get that for any $x_0 \in \Omega_0$, $\sum_{k=1}^{J(x_0)-1} \Lambda_{x_0,k} X_{x_0,k} = 0$. This gives that $(\Lambda_{x_0,k})_{k \in [1,J(x_0)-1]}$ is in the kernel of the matrix $(\sin(\ell k \pi / (J(x_0) - 1)))_{1 \leq j \leq J(x_0)-1}$. Since the sin discrete transform is invertible we obtain that for any $x_0 \in \Omega_0$ and $k \in [1, J(x_0) - 1]$, $\Lambda_{x_0,k} = 0$. Thus the family $X_{x_0,k}$ is a basis of eigenvectors.

7. Acknowledgements. The authors would like to thank Denis Gratias for the crystallography images, Jérémy Anger for some of natural images, Axel Davy who provided an OpenCL implementation of the NL-means algorithm and Thibaud Ehret for its insights and comments on denoising algorithms.

Appendix A. Eigenvalues.

Proof of Proposition 5. We fix $t \neq 0$ with $\|t\|_\infty < p$ and denote $C = C_t$. Without loss of generality we consider that $t_x > 0$ and $t_y > 0$. We consider $X$ an eigenvector of $C$. Let $\Omega_0 = (\Omega - t) \cap \Omega^c$ and the function $J : \Omega_0 \rightarrow [2, +\infty[$ such that for any $x_0 \in \Omega_0$

$$
J(x_0) = \arg\min \{k \in \mathbb{N} \setminus \{0\}, \ x_0 + kt \notin \Omega \}.
$$

It is clear that $I = \{(k,m), k \in [1, m - 1], m \in J(\Omega_0)\}$ is in bijection with $\Omega$. Let $x_0 \in \Omega_0$, $m = J(x_0)$ and $k \in [1, m - 1]$. We define $X_{x_0,k}$ over $\mathbb{Z}^2$ such that

$$
X_{x_0,k}(x_0 + k \ell t) = \sin \left( \frac{\ell k \pi}{m} \right) \text{ for } \ell \in [1, m - 1], \quad 0 \text{ otherwise}.
$$

Using that $\sin(a + b) + \sin(a - b) = 2 \sin(a) \cos(b)$, we have for any $x \in \mathbb{Z}^2$

$$
X_{x_0,k}(x + t) - 2 \cos \left( \frac{k\pi}{m} \right) X_{x_0,k}(x) + X_{x_0,k}(x - t) = 0.
$$

This implies that for any $x \in \mathbb{Z}^2$

$$
2X_{x_0,k}(x) - X_{x_0,k}(x + t) - X_{x_0,k}(x - t) = \left[ 2 - 2 \cos \left( \frac{k\pi}{m} \right) \right] X_{x_0,k}(x) = 4 \sin^2 \left( \frac{k\pi}{m} \right) X_{x_0,k}(x).
$$

Thus the one-dimensional vector (given by the raster-scan order on the $x$-axis) of the restriction of $X_{x_0,k}$ is an eigenvector of $C$ associated with eigenvalue $4 \sin^2 \left( \frac{k\pi}{m} \right)$. We show that this family of vectors is linearly-independent. Let $\Lambda_{x_0,k} \in \mathbb{R}$ such that

$$
\sum_{x_0 \in \Omega_0} \sum_{k=1}^{J(x_0)-1} \Lambda_{x_0,k} X_{x_0,k} = 0.
$$

Since $X_{x_0,k}$ and $X_{y_0,k'}$ have different support if and only if $x_0 \neq y_0$ we get that for any $x_0 \in \Omega_0$, $\sum_{k=1}^{J(x_0)-1} \Lambda_{x_0,k} X_{x_0,k} = 0$. This gives that $(\Lambda_{x_0,k})_{k \in [1,J(x_0)-1]}$ is in the kernel of the matrix $(\sin(\ell k \pi / (J(x_0) - 1)))_{1 \leq j \leq J(x_0)-1}$. Since the sin discrete transform is invertible we obtain that for any $x_0 \in \Omega_0$ and $k \in [1, J(x_0) - 1]$, $\Lambda_{x_0,k} = 0$. Thus the family $X_{x_0,k}$ is a basis of eigenvectors.
We aim at computing the cardinality of $K_{k,m} = \{X_{x_0,k}, J(x_0) = m\}$. By definition, in Proposition 5, $r_{k,m} = |K_{k,m}|$. First note that $|K_{k',m}| = |K_{k,m}|$. We give the following decomposition $\Omega_0 = \Omega_x \cup \Omega_y \cup \Omega_{x,y}$ with

$$\Omega_x = [-t_x, -1] \times [0, p-1-t_y], \quad \Omega_y = [0, p-1-t_x] \times [-t_y, -1], \quad \Omega_{x,y} = [-t_x, -1] \times [-t_y, -1].$$

Note that for all $x_0 \in \Omega_0$ we have that $x_0 + (q+1)t \notin \Omega$, with $q = \left\lfloor \frac{p}{|t_x| + |t_y|} \right\rfloor$. Thus $J(\Omega_0) \subset \llbracket 2, q+1 \rrbracket$. Let $m \in \llbracket 2, q-1 \rrbracket$. The cardinality of $K_{k,m}$ is the cardinality of $J^{-1}(m)$. Let $x_0 \in \Omega_x$ we have

$$x_0 = (i_0, j_0) \in K_{k,m} \iff \begin{cases} \quad i_0 + mt_x \geq p \\
\text{or} \quad j_0 + mt_y \geq p \\
\quad i_0 + (m - 1)t_x \leq p - 1 \\
\text{and} \quad j_0 + (m - 1)t_y \leq p - 1 
\end{cases}.$$

Since $x_0 \in \Omega_x$ we have $i_0 + mt_x \leq p - 1$, hence

$$x_0 = (i_0, j_0) \in K_{k,m} \iff j_0 + mt_y \geq p \text{ and } j_0 + (m - 1)t_y \leq p - 1.$$

Thus $\lvert \Omega_x \cap J^{-1}(m) \rvert = t_xt_y$. Similarly we get that $\lvert \Omega_y \cap J^{-1}(m) \rvert = t_xt_y$ and $\lvert \Omega_{x,y} \cap J^{-1}(m) \rvert = \emptyset$. Thus, $|K_{k,m}| = 2t_xt_y$.

We have computed $|K_{k,m}|$ for every $m \in \llbracket 2, q-1 \rrbracket$. In order to complete our study it only remains to compute $|K_{k,q+1}|$, since $|K_{k,q}|$ can be deduced from the summability condition and from $|K_{k,m}| = |K_{k',m}|$. We only compute $|K_{k,q+1}|$. We remark that $\Omega_x \cap J^{-1}(q + 1) = \Omega_y \cap J^{-1}(q + 1) = \emptyset$. Let $x_0 \in \Omega_{x,y}$ then $x_0 = -t + (x, y)$ with $x \in \llbracket 0, t_x - 1 \rrbracket$ and $y \in \llbracket 0, t_y - 1 \rrbracket$. We obtain the following equivalence

$$x_0 \in J^{-1}(q + 1) \iff \begin{cases} \quad -t_x + x + (q + 1)t_x \geq p \\
\quad -t_y + y + (q + 1)t_y \geq p \\
\text{or} \quad -t_x + x + qt_x \leq p - 1 \\
\text{and} \quad -t_y + y + qt_y \leq p - 1 
\end{cases}.$$

Since $qt_x \geq p$ or $qt_y \geq p$ we obtain that the first condition is always satisfied. Thus we get

$$x_0 \in J^{-1}(q + 1) \iff x \leq p - 1 - (q - 1)t_x \text{ and } y \leq p - 1 - (q - 1)t_y.$$

Using that $p - 1 - (q - 1)t_x = \left(\left\lfloor \frac{p}{t_x} \right\rfloor - q\right)t_x + t_x - 1 - p_x$, we conclude the proof.

**Appendix B. Update rules.** We derive the proof of Proposition 7.

**Proof.** Computing the minimum of $q(B, M|E)$ for fixed $B \in \mathbb{R}^d$, respectively fixed $M \in \mathbb{R}^d$. Let $q(B, M|E)$ be the function that we want to minimize, and let $\delta B$ and $\delta M$ be small perturbations of $B$ and $M$, respectively.
\( R^{2|E|} \), gives the update rule for \( M \), respectively for \( B \). We obtain that

\[
q(B, M|E) = \sum_{e \in E} m_e^2 \|b_1\|^2 + \sum_{e \in E} n_e^2 \|b_2\|^2 + 2 \sum_{e \in E} m_e n_e \langle b_1, b_2 \rangle - 2 \sum_{e \in E} m_e \langle b_1, e \rangle - 2 \sum_{e \in E} n_e \langle b_2, e \rangle + r(B, M) \\
= B^T (\Lambda_M \otimes \text{Id}_2) B - 2 \langle B, E_M \rangle + \alpha(M) \\
= \| (\Lambda_M \otimes \text{Id}_2)^{\frac{1}{2}} B - (\Lambda_M \otimes \text{Id}_2)^{-\frac{1}{2}} E_M \|^2 + \alpha(M) \\
= \| (\Lambda_M \otimes \text{Id}_2)^{\frac{1}{2}} \left( B - (\Lambda_M \otimes \text{Id}_2)^{-\frac{1}{2}} E_M \right) \|^2 + \alpha(M),
\]

where \( \alpha(M) \) depends only on \( M \). Similar derivation goes for \( B \) and we obtain the proposed update rules.

\section*{References}


[43] H. Lin, L. Wang, and S. Yang, Extracting periodicity of a regular texture based on autocorrelation


[65] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, Probabilistic non-local means, IEEE Signal

