



HAL
open science

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*, 2020, 10.1137/18M1228219 . hal-03037094

HAL Id: hal-03037094

<https://hal.science/hal-03037094>

Submitted on 2 Dec 2020

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

A PREPRINT

Valentin De Bortoli

CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

Agnès Desolneux

CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

Bruno Galerne

Institut Denis Poisson, Université d'Orléans, Université de Tours, CNRS

Arthur leclaire

Univ. Bordeaux, IMB, Bordeaux INP, CNRS, UMR 5251, F-33400 Talence, France.

December 2, 2020

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 and thus we give some examples of functions (auto-similarity and template similarity) which, given one or two images, computes a similarity measurement between patches. Two patches are said to be similar if the similarity measurement is small enough. 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 similarity 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 introduce a fast algorithm in order to assess redundancy in natural images and present applications in denoising, periodicity analysis and texture ranking.

()

Keywords patch, redundancy, statistical framework, *a contrario* method, image denoising, texture, periodicity analysis.

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 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 [64, 65, 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 $\Omega = \llbracket 0, M - 1 \rrbracket^2 \subset \mathbb{Z}^2$, with $M \in \mathbb{N} \setminus \{0\}$. Let $\omega \subset \mathbb{Z}^2$ be a patch domain. We introduce $P_\omega(u) = (\dot{u}(\mathbf{y}))_{\mathbf{y} \in \omega}$ the patch at position ω in the periodic extension of u to \mathbb{Z}^2 , denoted by \dot{u} . We define the auto-similarity with patch domain ω and offset $\mathbf{t} \in \mathbb{Z}^2$ by*

$$\mathcal{AS}(u, \mathbf{t}, \omega) = \|P_{\mathbf{t}+\omega}(u) - P_\omega(u)\|_2^2. \quad (1)$$

The auto-similarity computes the distance between a patch of u defined on a domain ω and the patch of u defined by the domain ω shifted by the offset vector \mathbf{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 $\Omega \subset \mathbb{Z}^2$ and a patch domain $\omega \subset \Omega$. 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 Ω . We will denote by \mathbb{P}_0 the probability distribution of U over \mathbb{R}^Ω , the images over Ω . We will assume that \mathbb{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: $\mathcal{AS}(u, \mathbf{t}, \omega) \leq a(\mathbf{t})$, where a , the threshold function, is defined over the offsets ($\mathbf{t} \in \mathbb{Z}^2$) but also depends on other parameters such as ω or \mathbb{P}_0 . The dependency of a with respect to \mathbf{t} cannot be omitted. For instance, even in a Gaussian white noise W , the probability distribution function of $\mathcal{AS}(W, \mathbf{t}, \omega)$ depends on \mathbf{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 \mathbb{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 \mathbf{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 $U \sim \mathbb{P}_0$, where \mathbb{P}_0 is a background microtexture model. We define the auto-similarity probability map AP for any $\mathbf{t} \in \Omega$, $\omega \subset \Omega$ and $a \in \mathbb{R}^\Omega$ by*

$$\text{AP}(\mathbf{t}, \omega, a) = \mathbb{P}_0[\mathcal{AS}(U, \mathbf{t}, \omega) \leq a(\mathbf{t})]. \quad (2)$$

We define the auto-similarity expected number of false alarms ANFA by

$$\text{ANFA}(\omega, a) = \sum_{\mathbf{t} \in \Omega} \text{AP}(\mathbf{t}, \omega, a). \quad (3)$$

Note that $\text{AP}(\mathbf{t}, \omega, a)$ corresponds to the probability that $\omega + \mathbf{t}$ is similar to ω in the background model U . For any $\mathbf{t} \in \Omega$, the cumulative distribution function of the auto-similarity random variable $\mathcal{AS}(U, \mathbf{t}, \omega)$ under \mathbb{P}_0 evaluated at value $\alpha(\mathbf{t})$ is given by $\text{AP}(\mathbf{t}, \omega, \alpha(\mathbf{t}))$. We denote by $q \mapsto \text{AP}^{-1}(\mathbf{t}, \omega, q)$ the inverse cumulative distribution function, potentially defined by a generalized inverse ($\text{AP}^{-1}(\mathbf{t}, \omega, q) = \inf\{\alpha(\mathbf{t}) \in \mathbb{R}, \text{AP}(\mathbf{t}, \omega, \alpha(\mathbf{t})) \geq q\}$), of the auto-similarity random variable for a fixed offset \mathbf{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 \mathbf{t} is said to be detected with respect to a , if $\mathcal{AS}(u, \mathbf{t}, \omega) \leq a(\mathbf{t})$.

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 $\mathcal{AS}(U, \mathbf{t}, \omega)$ is invertible for every $\mathbf{t} \in \Omega$. This ensures that for any $\mathbf{t} \in \Omega$ and $q \in (0, 1)$ we have

$$\text{AP}(\mathbf{t}, \omega, \text{AP}^{-1}(\mathbf{t}, \omega, q)) = q. \quad (4)$$

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

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

Proof: Using (3), and $a(\mathbf{t}) = \text{AP}^{-1}(\mathbf{t}, \omega, \text{NFA}_{\max}/|\Omega|)$, we get

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

where the last equality is obtained using (4). Concerning the upper-bound, we have, using the Markov inequality and (2), for any $n \in \mathbb{N} \setminus \{0\}$

$$\begin{aligned} \mathbb{P}_0[\text{“at least } n \text{ offsets are detected in } U\text{”}] &= \mathbb{P}_0\left[\sum_{\mathbf{t} \in \Omega} \mathbb{1}_{\mathcal{AS}(U, \mathbf{t}, \omega) \leq a(\mathbf{t})} \geq n\right] \\ &\leq \frac{\sum_{\mathbf{t} \in \Omega} \mathbb{E}[\mathbb{1}_{\mathcal{AS}(U, \mathbf{t}, \omega) \leq a(\mathbf{t})}]}{n} \leq \frac{\text{NFA}_{\max}}{n}, \end{aligned}$$

where $\mathbb{1}_{\mathcal{AS}(U, \mathbf{t}, \omega) \leq a(\mathbf{t})} = 1$ if $\mathcal{AS}(U, \mathbf{t}, \omega) \leq a(\mathbf{t})$ and 0 otherwise. \square

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

$$\mathcal{AS}(u, \mathbf{t}, \omega) \leq \text{AP}^{-1}(\mathbf{t}, \omega, \text{NFA}_{\max}/|\Omega|). \quad (5)$$

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 (5), we obtain that an offset is detected if and only if

$$\mathbb{P}_0[\mathcal{AS}(U, \mathbf{t}, \omega) \leq \mathcal{AS}(u, \mathbf{t}, \omega)] = \text{AP}(\mathbf{t}, \omega, \mathcal{AS}(u, \mathbf{t}, \omega)) \leq \text{NFA}_{\max}/|\Omega|. \quad (6)$$

Therefore, the thresholding operation can be conducted either on $\mathcal{AS}(u, \mathbf{t}, \omega)$, see (5), or on $\text{AP}(\mathbf{t}, \omega, \mathcal{AS}(u, \mathbf{t}, \omega))$, see (6). This property will be used in Section 3.2 to define a similarity detection algorithm based on the evaluation of $\mathcal{AS}(u, \mathbf{t}, \omega)$.

3 Gaussian model and detection algorithm

3.1 Choice of background model

In this section we compute $\text{AP}(\mathbf{t}, \omega, \alpha)$, *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 over \mathbb{R}^Ω 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, $U = f * W$, where $*$ is the periodic convolution operator over Ω given by $v * w(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega} v(\mathbf{y})w(\mathbf{x} - \mathbf{y})$ and W is a white noise over Ω , i.e. $(W(\mathbf{x}))_{\mathbf{x} \in \Omega}$ 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

$$m_u = \sum_{\mathbf{x} \in \Omega} u(\mathbf{x})/|\Omega|, \quad \text{and} \quad U = |\Omega|^{-1/2}(u - m_u) * W. \quad (7)$$

Note that if U is given by (7) we have for any $\mathbf{x}, \mathbf{y} \in \Omega$

$$\mathbb{E}[U(\mathbf{x})] = 0 \quad \text{and} \quad \text{Cov}[U(\mathbf{x}), U(\mathbf{y})] = |\Omega|^{-1} \sum_{\mathbf{z} \in \Omega} (\dot{u}(\mathbf{z}) - m_u)(\dot{u}(\mathbf{z} - (\mathbf{y} - \mathbf{x})) - m_u). \quad (8)$$

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

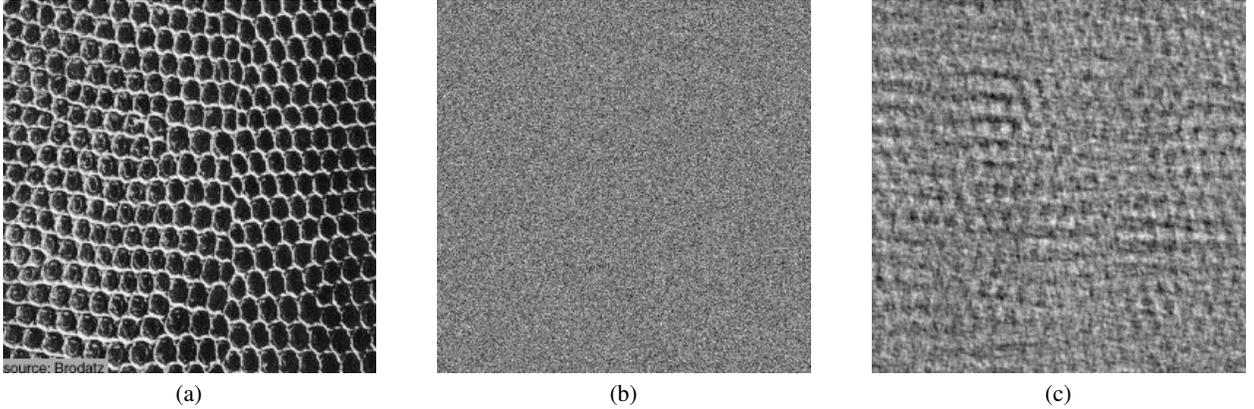


Figure 1: **Examples of microtexture models.** In (a) we present an original 256×256 image. In (b) and (c) we derive two microtexture models. In (b) we present a Gaussian white noise and in (c) the microtexture model given by (7). Note that (c) shows more local structure than (b).

3.2 Detection algorithm

In this section, Ω is a finite square domain in \mathbb{Z}^2 . We fix $\omega \subset \Omega$. We also define f , a function over Ω . We consider the Gaussian random field $U = f * W$, where W is a Gaussian white noise over Ω . We denote by Γ_f the autocorrelation of f , i.e. $\Gamma_f = f * \check{f}$ where for any $\mathbf{x} \in \Omega$, $\check{f}(\mathbf{x}) = f(-\mathbf{x})$. We introduce the offset correlation function Δ_f defined for any $\mathbf{t}, \mathbf{x} \in \Omega$ by

$$\Delta_f(\mathbf{t}, \mathbf{x}) = 2\Gamma_f(\mathbf{x}) - \Gamma_f(\mathbf{x} + \mathbf{t}) - \Gamma_f(\mathbf{x} - \mathbf{t}). \quad (9)$$

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

Proposition 2 (Squared ℓ^2 auto-similarity function exact probability distribution function) Let $\Omega = \llbracket 0, M-1 \rrbracket^2$ 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 Ω . Then, for any $\mathbf{t} \in \Omega$, $\mathcal{AS}(U, \mathbf{t}, \omega)$ has the same distribution as $\sum_{k=0}^{|\omega|-1} \lambda_k(\mathbf{t}, \omega) Z_k$, with Z_k independent chi-square random variables with parameter 1 and $\lambda_k(\mathbf{t}, \omega)$ the eigenvalues of the covariance matrix $C_{\mathbf{t}}$ associated with function $\Delta_f(\mathbf{t}, \cdot)$ restricted to ω , defined in (9), i.e for any $\mathbf{x}_1, \mathbf{x}_2 \in \omega$, $C_{\mathbf{t}}(\mathbf{x}_1, \mathbf{x}_2) = \Delta_f(\mathbf{t}, \mathbf{x}_1 - \mathbf{x}_2)$.

As a consequence if $f = \delta_0$, i.e. U is a Gaussian white noise, and $\{\mathbf{x} + \mathbf{t}, \mathbf{x} \in \omega\} \cap \omega = \emptyset$, i.e. there is no overlapping between the patch domain ω and its shifted version, then $\mathcal{AS}(U, \mathbf{t}, \omega)$ 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 $\lambda_k(\mathbf{t}, \omega)$ 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

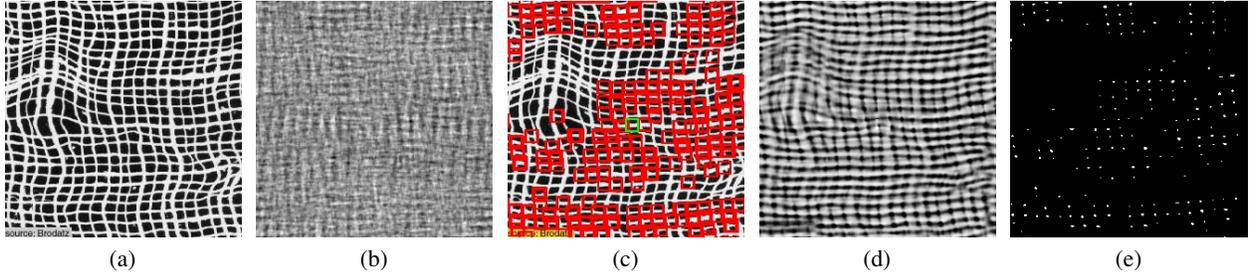


Figure 2: **Outputs of Algorithm 1.** In (a) we present an original 256×256 image. In (b) we present the associated microtexture model given by (7). In (c) the green patch is the input patch, *i.e.* $P_\omega(u)$. In this experiment NFA_{\max} is set to 1. In (d), respectively (e), we present the output P_{map} , respectively D_{map} , of Algorithm 1. In (c) we show in red the patches corresponding to the identified offsets in P_{map} .

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 $\mathcal{AS}(U, \mathbf{t}, \omega)$ and we refer to [15] for further details.

In Algorithm 1 we propose an *a contrario* framework for spatial redundancy detection. We suppose that u and ω are provided by the user. Using Proposition 1 and (6), we say that an offset is detected if $AP(\mathbf{t}, \omega, \mathcal{AS}(u, \mathbf{t}, \omega)) \leq NFA_{\max}/|\Omega|$. The value NFA_{\max} is supposed to be set by the user. The background model used in the auto-similarity detection is the one given in (7). Therefore, Proposition 2 and the discussion that follows can be used to compute an approximation of $AP(\mathbf{t}, \omega, \mathcal{AS}(u, \mathbf{t}, \omega))$. In Figure 2 we apply Algorithm 1 to a texture image.

Algorithm 1 Auto-similarity detection

```

1: function AUTOSIM-DETECTION( $u, \omega, NFA_{\max}$ )
2:   for  $\mathbf{t} \in \Omega$  do
3:      $\text{val} \leftarrow \mathcal{AS}(u, \mathbf{t}, \omega)$ 
4:      $P_{map}(\mathbf{t}) \leftarrow AP(\mathbf{t}, \omega, \text{val})$  ▷  $AP(\mathbf{t}, \omega, \text{val})$  approximation detailed in Section 3.2
5:      $D_{map}(\mathbf{t}) \leftarrow \mathbb{1}_{P_{map}(\mathbf{t}) \leq NFA_{\max}/|\Omega|}$ 
6:   end for
7:   return the images  $P_{map}, D_{map}$ 
8: end function

```

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 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, 22] 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 [23, 41] or to take into account the underlying geometry of the patch space [34]. Among the state-of-the-art denoising algorithms, see [39] 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 χ_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 2. In what follows we fix a “clean”, or original, image u_0 defined over Ω , 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 σ is known. Note that there exist several algorithms to estimate σ 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 ω_0 be a centered 8×8 patch domain. For a patch window $\omega = \mathbf{x} + \omega_0$ the patch search window T will be defined by

$$T = \{\mathbf{t} \in \mathbb{Z}^2, \mathbf{t} + \omega \subset \Omega, \|\mathbf{t}\|_\infty \leq c\}, \quad (10)$$

with $c \in \mathbb{N}$. $|T|$ denotes the cardinality of T . There exists a large literature concerning the setting of c and ω_0 , see [22]. 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 ω , we obtain a pixel at position \mathbf{x} in the denoised image \hat{u} using the following average, see [6],

$$\hat{u}(\mathbf{x}) = |\{\mathbf{t} \in \Omega, \text{s.t } \mathbf{x} \in \mathbf{t} + \omega \subset \Omega\}|^{-1} \sum_{\mathbf{t} \in \Omega, \text{s.t } \mathbf{x} \in \mathbf{t} + \omega \subset \Omega} \hat{p}(u, \mathbf{t} + \omega)(\mathbf{x}). \quad (11)$$

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 2 NL-means threshold

```

1: function NL-MEANS-THRESHOLD( $u, \sigma, \omega_0, c, a$ )
2:   for  $\mathbf{x} \in \mathbb{Z}^2, \mathbf{x} + \omega_0 \subset \Omega$  do
3:      $\omega \leftarrow \mathbf{x} + \omega_0$ 
4:      $T \leftarrow$  defined by (10)
5:      $N_\omega(u) \leftarrow 0$ 
6:      $\hat{p}(u, \omega) \leftarrow 0$ 
7:     for  $\mathbf{t} \in T$  do
8:       if  $\mathcal{AS}(u, \mathbf{t}, \omega) \leq \sigma^2 a(\mathbf{t})$  then  $\triangleright$  always true for  $\mathbf{t} = 0$ 
9:          $\hat{p}(u, \omega) \leftarrow \frac{N_\omega(u)}{N_\omega(u)+1} \hat{p}(u, \omega) + \frac{1}{N_\omega(u)+1} P_{\mathbf{t}+\omega}(u)$   $\triangleright P_\omega(u)$  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 (11)
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

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

In the original NL-means method, we have

$$\lambda_{\mathbf{t}} = \frac{\exp\left(-\frac{\mathcal{AS}(u, \mathbf{t}, \omega)}{h^2}\right)}{\sum_{\mathbf{t} \in T} \exp\left(-\frac{\mathcal{AS}(u, \mathbf{t}, \omega)}{h^2}\right)}. \quad (12)$$

Setting h is not trivial and depends on many parameters (patch size, search window size, content of the original image). As in Algorithm 2, we denote $N_\omega(u) = \sum_{\mathbf{t} \in T} \mathbb{1}_{\mathcal{AS}(u, \mathbf{t}, \omega) \leq a(\mathbf{t})}$. The following proposition, similar to Proposition 1, gives a method for setting a . We say that an offset \mathbf{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 (10) and let $a \in \mathbb{R}^\Omega$ be defined for any $\mathbf{t} \in \Omega$ by

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

with background model being a Gaussian white noise W , i.e. $f = \delta_0$ in Definition 4. Let T be defined in (10) and $N_\omega(W) \in \{0, \dots, T\}$ the random number of selected patches used to denoise the patch $P_\omega(W)$, see Algorithm 2. Then for any $n \in \mathbb{N} \setminus \{0\}$ 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

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

□

In this 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 Ω , $u_0 \in \mathbb{R}^\Omega$ and $\sigma > 0$. Let $\mathbf{x} \in \Omega$ and $\omega = \mathbf{x} + \omega_0$ be a fixed patch and let $\text{NFA}_{\max} \in [0, |T|]$. We introduce the random set $\hat{T} = \{\mathbf{t} \in T, \mathcal{AS}(U, \mathbf{t}, \omega) \leq \sigma^2 a(\mathbf{t})\}$ (the selected offsets) with $a(\mathbf{t}) = \text{AP}^{-1}(\mathbf{t}, \omega, 1 - \text{NFA}_{\max}/|T|)$ as in Proposition 3 and T defined in (10). Let $a_T = \max_{\mathbf{t} \in T} a(\mathbf{t})$. Then for any $a_W > 0$, setting $\varepsilon_W = 1 - \mathbb{P} [\|P_\omega(W)\|_2^2 \leq a_W \mid \hat{T}]$, we have

$$\mathbb{P} [\|\hat{p}(U, \omega) - P_\omega(u_0)\|_2 \leq \sigma(a_T^{1/2} + a_W^{1/2}) \mid \hat{T}] \geq 1 - \varepsilon_W . \quad (13)$$

Proof: We have for any $\mathbf{t} \in \hat{T}$

$$\begin{aligned} \|P_{\mathbf{t}+\omega}(U) - P_\omega(u_0)\|_2 &\leq \|P_{\mathbf{t}+\omega}(U) - P_\omega(U) + P_\omega(U) - P_\omega(u_0)\|_2 \\ &\leq \|P_{\mathbf{t}+\omega}(U) - P_\omega(U)\|_2 + \|P_\omega(U) - P_\omega(u_0)\|_2 \\ &\leq \sigma a_T^{1/2} + \sigma \|P_\omega(W)\|_2 . \end{aligned}$$

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

$$\left\{ \|P_\omega(W)\|_2 \leq a_W^{1/2} \right\} \subset \left\{ \|P_{\mathbf{t}+\omega}(U) - P_\omega(u_0)\|_2 \leq \sigma(a_T^{1/2} + a_W^{1/2}) \right\} ,$$

We also have that by definition of ε_W

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

□

In our applications we use Algorithm 2 with $a(\mathbf{t}) = \text{AP}^{-1}(\mathbf{t}, \omega, 1 - \text{NFA}_{\max}/|T|)$. Therefore we need to compute $a(\mathbf{t}) = \text{AP}^{-1}(\mathbf{t}, \omega, 1 - \text{NFA}_{\max}/|T|)$ 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 $\mathbf{t} = (t_x, t_y) \in \mathbb{Z}^2 \setminus \{0\}$, $C_{\mathbf{t}}$ as in Proposition 2 with $f = \delta_0$ and $\omega = \llbracket 0, p-1 \rrbracket^2$, with $p \in \mathbb{N}$. We have, expressing $C_{\mathbf{t}}$ in the basis corresponding to the raster scan order on the x -axis

$$C_{\mathbf{t}} = \begin{pmatrix} B_0 & B_1 & \dots & B_{p-1} \\ B_1^\top & B_0 & \ddots & \vdots \\ \vdots & \ddots & B_0 & B_1 \\ B_{p-1}^\top & \dots & B_1^\top & B_0 \end{pmatrix} + 2\text{Id} , \quad \begin{cases} B_\ell = D_{|t_y|} \in \mathcal{M}_p(\mathbb{R}) & \text{if } \ell = |t_x| \\ B_\ell = 0 & \text{otherwise} \end{cases}$$

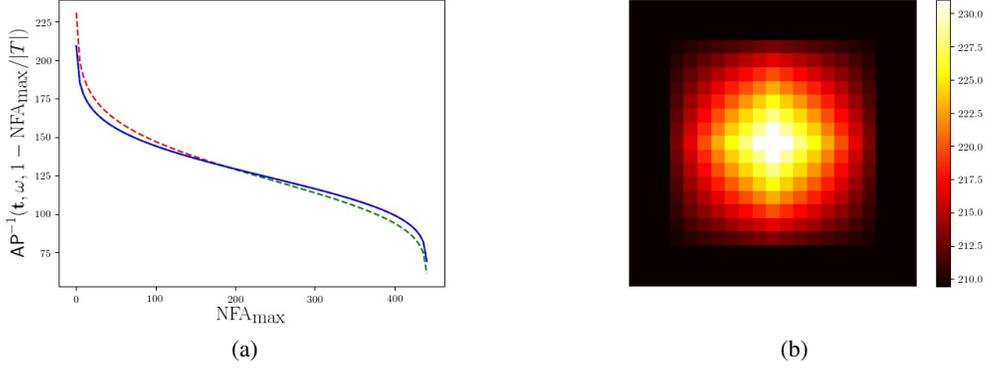


Figure 3: **Thresholds dependency in NFA_{max} .** In (a) we display $a(\mathbf{t}) = AP^{-1}(\mathbf{t}, \omega, 1 - NFA_{max}/|T|)$ as a function of NFA_{max} . The patch size is fixed to 8×8 and the offsets \mathbf{t} satisfy $\|\mathbf{t}\|_{\infty} \leq 10$, hence $|T| = 21^2 = 441$. The red dashed line is given by $\max_{\mathbf{t} \in T} a(\mathbf{t})$ and the green dashed line by $\min_{\mathbf{t} \in T} a(\mathbf{t})$. 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 $a(\mathbf{t})$ and the minimum of $a(\mathbf{t})$ is of 13.0%. In (b) we display the mapping $\mathbf{t} \mapsto a(\mathbf{t})$ for $NFA_{max} = 0.5$, the central pixel corresponds to $\mathbf{t} = \mathbf{0}$. Note that $a(\mathbf{t})$ decreases as $\|\mathbf{t}\|$ increases and is constant when, $\|\mathbf{t}\|_{\infty} \geq 8$.

where D_j is a zero matrix with ones on the j -th diagonal. The eigenvalues of $C_{\mathbf{t}}$ are given by $\lambda_{m,k} = 4 \sin^2\left(\frac{k\pi}{2m}\right)$ with multiplicity $r_{m,k}$ where $m \in \llbracket 2, q+1 \rrbracket$, $k \in \llbracket 1, m-1 \rrbracket$ and $q = \lceil \frac{p}{|t_x| \vee |t_y|} \rceil$. 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 $r_x = \left(\lceil \frac{p}{|t_x|} \rceil - q\right) |t_x| + |t_x| - p_x$, where $p_x = |t_x| \lceil \frac{p}{|t_x|} \rceil - p$. We define r_y in the same manner. A similar proposition holds if $t_y \neq 0$.

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 $a(\mathbf{t})$ for fixed patch size (8×8) and patch search window (21×21). Thus in our implementation we suppose that $a(\mathbf{t}) = AP^{-1}(\mathbf{t}, \omega, 1 - NFA_{max}/|T|)$ is constant and set its value to the mean of $a(\mathbf{t})$ over $\mathbf{t} \in T$.

4.2 Some experimental results

In the following paragraph we present and comment some results of our threshold NL-means algorithm, see Algorithm 2. We recall that we use $a(\mathbf{t}) = \sum_{\mathbf{t} \in T} AP^{-1}(\mathbf{t}, \omega, 1 - NFA_{max}/|T|)/|T|$. In Figure 4 we present a first comparison with the NL-means algorithm. Perceptual results as well as Peak Signal to Noise Ratio (PSNR) measurements¹ 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 $a(\mathbf{t})$ is high, *i.e.* $NFA_{max} \ll |T|$ 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,

¹PSNR(u, v) = $10 \log_{10} \left(\frac{\max_{\Omega} u^2}{\|u-v\|_2^2} \right)$.

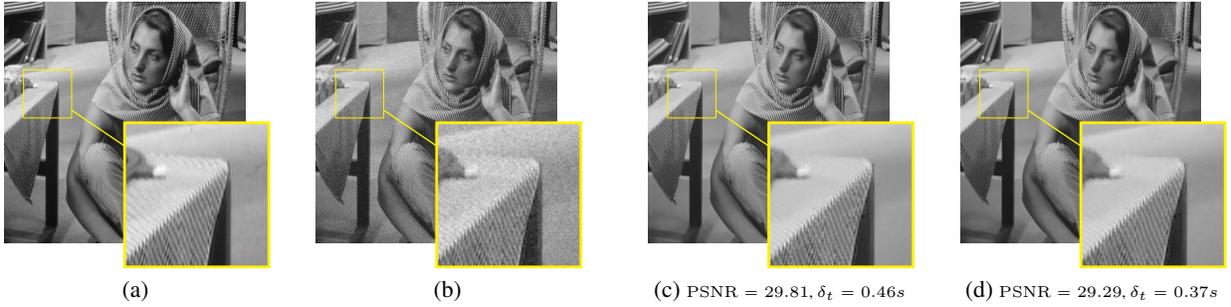


Figure 4: **Visual results.** In (a) we present an original image (Barbara) scaled between 0 and 255. In (b) we add Gaussian white noise with $\sigma = 10$. We recall that the patch domain is fixed to ω_0 being a 8×8 square. In (c) we present the denoised results with NL-means threshold, Algorithm 2, where $NFA_{\max} = 4.41$, 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), δ_t is the running time of the algorithm. We can observe that our algorithm is slightly slower than NL-means.

in this case details are lost due to the averaging process. By setting a conservative threshold, *e.g.* $NFA_{\max}/|T| \approx 0.1$, 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 2 performs better than the original NL-means algorithm. By setting $NFA_{\max}/|T| = 0.01$ 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 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 used the quantization of images, relying on co-occurrence matrices and statistical tools like χ_2 tests or κ 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 $\mathbf{t} \in \Omega$ by $\mathcal{I}(\mathbf{t}) = \sum_{(i,j) \in [0, N_g]^2} (i - j)^2 \left(\sum_{\mathbf{z} \in \Omega} \mathbb{1}_{\hat{u}(\mathbf{z})=i} \mathbb{1}_{\hat{u}(\mathbf{z}+\mathbf{t})=j} \right)$, 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 ω -inertia for any $\mathbf{t} \in \Omega$ by $\mathcal{I}_\omega(\mathbf{t}) = \sum_{(i,j) \in [0, N_g]^2} (i - j)^2 \left(\sum_{\mathbf{z} \in \omega} \mathbb{1}_{\hat{u}(\mathbf{z})=i} \mathbb{1}_{\hat{u}(\mathbf{z}+\mathbf{t})=j} \right)$. 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 $\mathbf{x} \in \Omega$, $u(\mathbf{x}) \in [0, N_g]$. We have $\mathcal{I}_\omega(\mathbf{t}) = \mathcal{AS}(u, \mathbf{t}, \omega)$.*

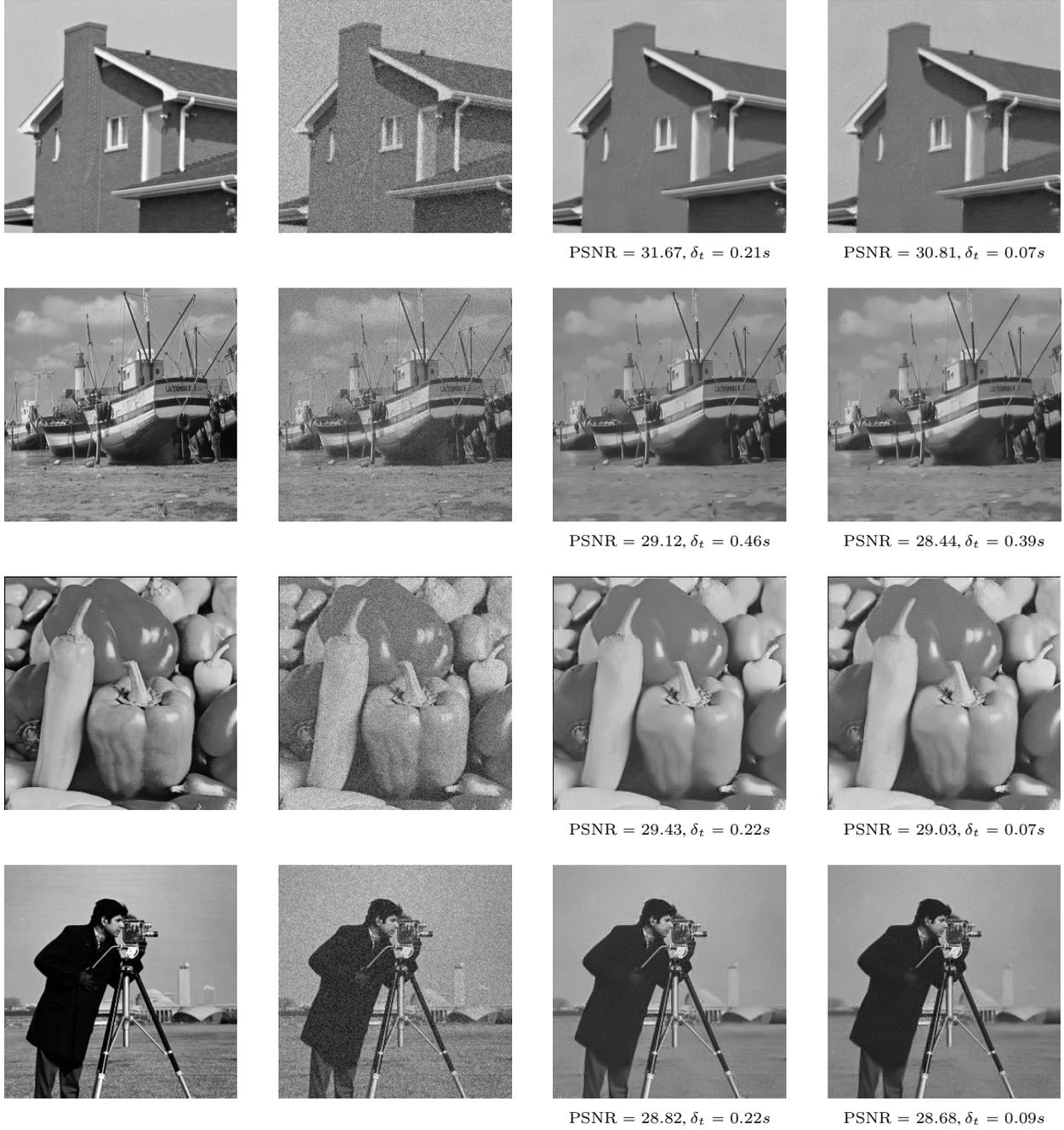


Figure 5: *NL-means comparison*. In this figure we compare Algorithm 2 with the traditional NL-means algorithm. Here ω_0 is fixed to be a 8×8 square. The first column contains clean images, the second column represents the same images corrupted by a Gaussian noise with $\sigma = 20$. The third column is the output of Algorithm 2 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 (12). Perceptual results and PSNR are comparable, even though our algorithm yields slightly better PSNR values. We also present the running times δ_t of both algorithm. Our algorithm is slower than NL-means as it computes the threshold before running the NL-means algorithm.

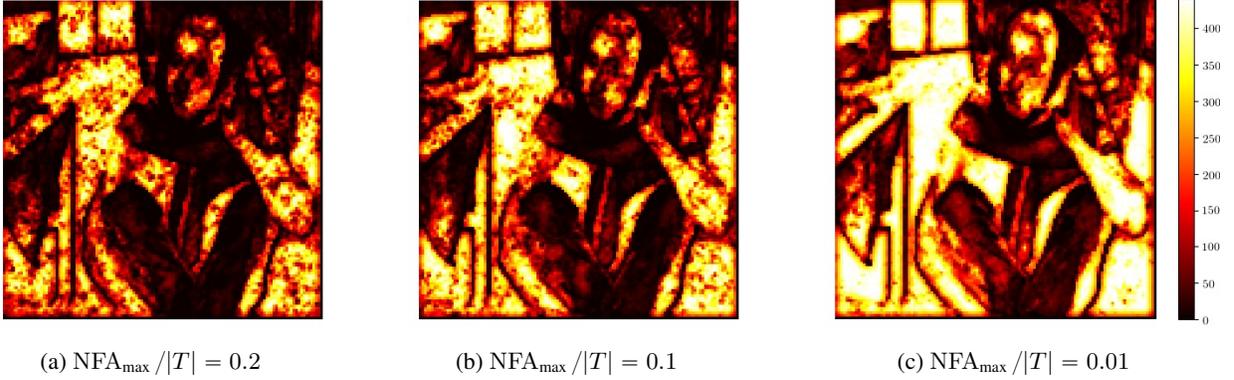


Figure 6: **Number of detections.** In this figure we present, for each denoised pixel, the number of detected offsets used to compute the denoised patch, *i.e.* the cardinality of \hat{T} , see Proposition 4. A white pixel means that the number of detected offsets is maximal and a black pixel means that the number of detected offsets is 1, *i.e.* the patch is not denoised. As NFA_{\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 $NFA_{\max}/|T| = 0.1$, pixels located in textured neighborhoods use in average 20 to 40 patches to perform denoising.



Figure 7: **Comparison with BM3D.** We compare Algorithm 2 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 [40]. The result in (c) corresponds to the output of Algorithm 2 with $NFA_{\max}/|T| = 0.01$. The output (c) is too blurry compared to (b). In order to correct this behavior we set $NFA_{\max}/|T| = 0.1$ 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 [23]. For instance, some patches in the scarf are not denoised anymore.

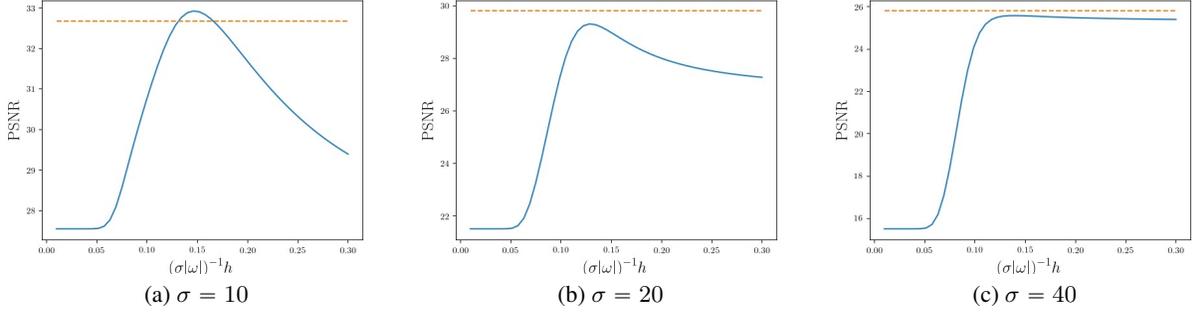


Figure 8: PSNR *study*. In this figure we present the evolution of the PSNR for different values of the parameter h of the original NL-means method, see (12), in blue, computed on the Barbara image. The x -axis represents $\frac{h}{\sigma|\omega|}$. The orange dashed line is the PSNR obtained for the threshold NL-means algorithm (Algorithm 2) with $\text{NFA}_{\max}/|T| = 0.01$. 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 .

Proof: For any $\mathbf{t} \in \Omega$ we have

$$\begin{aligned} \mathcal{I}_\omega(\mathbf{t}) &= \sum_{(i,j) \in \llbracket 0, N_g \rrbracket^2} (i-j)^2 \sum_{\mathbf{x} \in \omega} \mathbb{1}_{\dot{u}(\mathbf{x})=i} \mathbb{1}_{\dot{u}(\mathbf{x}+\mathbf{t})=j} = \sum_{\mathbf{x} \in \omega, (i,j) \in \llbracket 0, N_g \rrbracket^2} (i-j)^2 \mathbb{1}_{\dot{u}(\mathbf{x})=i} \mathbb{1}_{\dot{u}(\mathbf{x}+\mathbf{t})=j} \\ &= \sum_{\mathbf{x} \in \omega} (\dot{u}(\mathbf{x}) - \dot{u}(\mathbf{x} + \mathbf{t}))^2 = \mathcal{AS}(u, \mathbf{t}, \omega). \end{aligned}$$

□

If $\omega = \Omega$ then the ω -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 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 1) where the patch domain ω is fixed. Using these detections we build a graph with a few nodes (usually approximately 20 nodes for a 256×256 image). We use the same notation for the detection mapping $\mathbf{t} \mapsto \mathbb{1}_{\mathcal{AS}_i(u, \mathbf{t}, \omega) \leq a(\mathbf{t})}$ *i.e.* the D_{map} output of Algorithm 1, which is a binary function over the offsets, and the set of detected offsets. We recall that two pixel coordinates \mathbf{x} and \mathbf{y} are said to be 8-connected if $\mathbf{x} = \mathbf{y} + (\delta_x, \delta_y)$ 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_{map} we note \mathbf{v}_k one position for which the minimum of $\mathcal{AS}(u, \mathbf{t}, \omega)$ over \mathcal{C}_k is achieved. The set of vertices V is defined as $V = (\mathbf{v}_k)_{k \in \llbracket 1, N_{\mathcal{G}} \rrbracket}$ where $N_{\mathcal{G}}$ is the number of 8-connected components in D_{map} ;
- ▶ **Edges:** each vertex is linked with its four nearest neighbors in the sense of the Euclidean distance, defining four unoriented edges.

Referring 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 $\mathbf{e} \in E$, a couple of integers $(m_{\mathbf{e}}, n_{\mathbf{e}}) \in \mathbb{Z}^2$ such that \mathbf{e} has the same distribution as $m_{\mathbf{e}}b_1 + n_{\mathbf{e}}b_2 + \sigma Z_{\mathbf{e}}$, with $(Z_{\mathbf{e}})_{\mathbf{e} \in E}$ independent standard Gaussian random variables in \mathbb{R}^2 and $\sigma > 0$. We denote by M the vector of all coefficients $(m_{\mathbf{e}}, n_{\mathbf{e}})_{\mathbf{e} \in E} \in \mathbb{Z}^{2|E|}$.

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

$$\mathcal{L}(B, M, \sigma^2 | E) = -2(|E| + 1) \log(\sigma^2) - \frac{1}{2\sigma^2} \underbrace{\left(\sum_{\mathbf{e} \in E} \|m_{\mathbf{e}} b_1 + n_{\mathbf{e}} b_2 - \mathbf{e}\|^2 + r(B, M) \right)}_{q(B, M | E)}, \quad (14)$$

where $r(B, M) = \delta_B \|B\|_2^2 + \delta_M \|M\|_2^2$ with $\delta_B, \delta_M \geq 0$. A discussion on the dependence of the model on the hyperparameters (δ_B, δ_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 B and 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.

Algorithm 3 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 ▷ initialization discussed in the end of Section 5.2
4:   for  $n \leftarrow 0$  to  $N_{it} - 1$  do
5:      $\tilde{M} \leftarrow \operatorname{argmin}_{M \in \mathbb{R}^{2|E|}} q(B_n, M | E)$  ▷ expression given in Proposition 7
6:     if  $q(E | B_n, [\tilde{M}]) < q(E | B_n, M_n)$  then ▷  $[\cdot]$  is the nearest integer operator
7:        $M_{n+1} \leftarrow [\tilde{M}]$ 
8:     else
9:        $M_{n+1} \leftarrow M_n$ 
10:    end if
11:     $B_{n+1} \leftarrow \operatorname{argmin}_{B \in \mathbb{R}^4} q(B, M_{n+1} | E)$  ▷ expression given in Proposition 7
12:  end for
13:   $\sigma_{N_{it}}^2 \leftarrow \operatorname{argmin}_{\sigma^2 \in \mathbb{R}_+} -\mathcal{L}(B_{N_{it}}, M_{N_{it}}, \sigma^2 | E)$ 
14:  return  $B_{N_{it}}, M_{N_{it}}, \sigma_{N_{it}}^2$ 
15: end function

```

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

Proposition 7 (Alternate minimization update rule) *In Algorithm 3, we get for any $n \in \mathbb{N}$*

$$\tilde{M} = (\Lambda_{B_n} \otimes \operatorname{Id}_{|E|})^{-1} E_{B_n} \in \mathbb{R}^{2|E|}, \quad B_{n+1} = (\Lambda_{M_{n+1}} \otimes \operatorname{Id}_2)^{-1} E_{M_{n+1}} \in \mathbb{R}^4,$$

with \otimes the tensor product between matrices and

$$(a) \quad \Lambda_B = \begin{pmatrix} \|b_1\|^2 + \delta_B & \langle b_1, b_2 \rangle \\ \langle b_1, b_2 \rangle & \|b_2\|^2 + \delta_B \end{pmatrix}, \quad \Lambda_M = \begin{pmatrix} \|M_1\|^2 + \delta_M & \langle M_1, M_2 \rangle \\ \langle M_1, M_2 \rangle & \|M_2\|^2 + \delta_M \end{pmatrix};$$

$$(b) \quad E_B = \begin{pmatrix} \langle \mathbf{e}, b_1 \rangle_{\mathbf{e} \in E} \\ \langle \mathbf{e}, b_2 \rangle_{\mathbf{e} \in E} \end{pmatrix}, \quad E_M = \begin{pmatrix} \sum_{\mathbf{e} \in E} m_{\mathbf{e}} \mathbf{e} \\ \sum_{\mathbf{e} \in E} n_{\mathbf{e}} \mathbf{e} \end{pmatrix}.$$

Proof: The proof is postponed to Appendix B. □

Note that if B is orthogonal, i.e. $\langle b_1, b_2 \rangle = 0$ then Λ_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.*

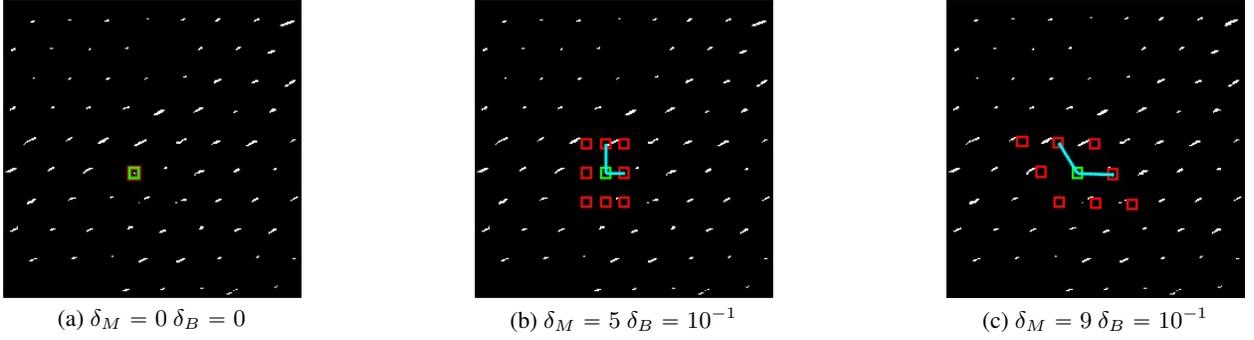


Figure 9: **Influence of hyperparameters.** In this experiment we assess the importance of the hyperparameters. We consider Algorithm 3 with input graph a detection map, output of Algorithm 1. The initialization in the three cases is the canonical basis $((0, 1), (1, 0))$. In (a), since the initial basis 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 3. Increasing δ_M up to 9, in (c) we obtain a perceptually correct lattice. For δ_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 δ_M is more flexible and that $\delta_M \in (1, 20)$ gives satisfying perceptual results if the initialization heuristics proposed in Section 5.2 is chosen.

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

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

Hence for every $n \geq n_0 + 1$, $\mathcal{L}_n(\sigma) = \mathcal{L}(B_n, M_n, \sigma^2 | E) = \tilde{\mathcal{L}}(\sigma)$. 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 of lines 6 and 7 of Algorithm 3) which is absurd. Thus $(M_n)_{n \in \mathbb{N}}$ is stationary and so is $(B_n)_{n \in \mathbb{N}}$. \square

In Algorithm 3 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 1 in order to extract auto-similarity given a patch in an original image u , which implies that the patch domain ω is set by the user. Recall that in Algorithm 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 3 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 \mathbb{P}_0 ,
- (b) NFA_{\max} parameter in Algorithm 1,
- (c) patch domain ω .

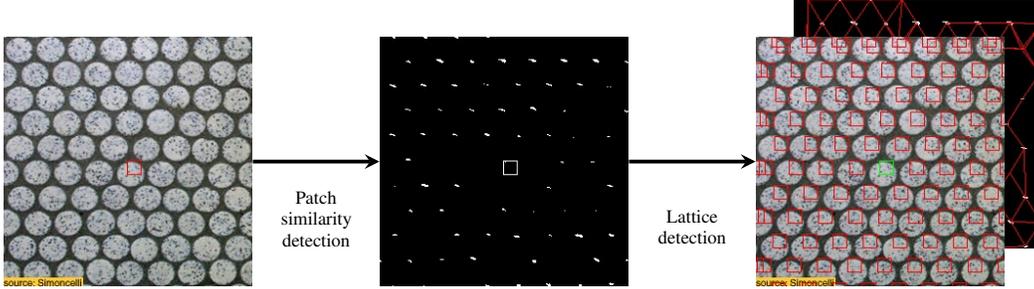


Figure 10: **Lattice proposal algorithm.** Lattice detection and extraction in images require a patch from the user and compute a binary image containing all the offsets with correct similarity as well as a lattice matching the underlying graph. The patch auto-similarity detection step was presented 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_{\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_{\max} must be set carefully otherwise two situations can occur:

- (a) if NFA_{\max} is too high, too many detections can be obtained (true perceptual detections are not differentiated from false positives) ;
- (b) if NFA_{\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_{\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 order to obtain the same detection map as Algorithm 1 one must carefully set the threshold parameter, NFA_{\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 (14).

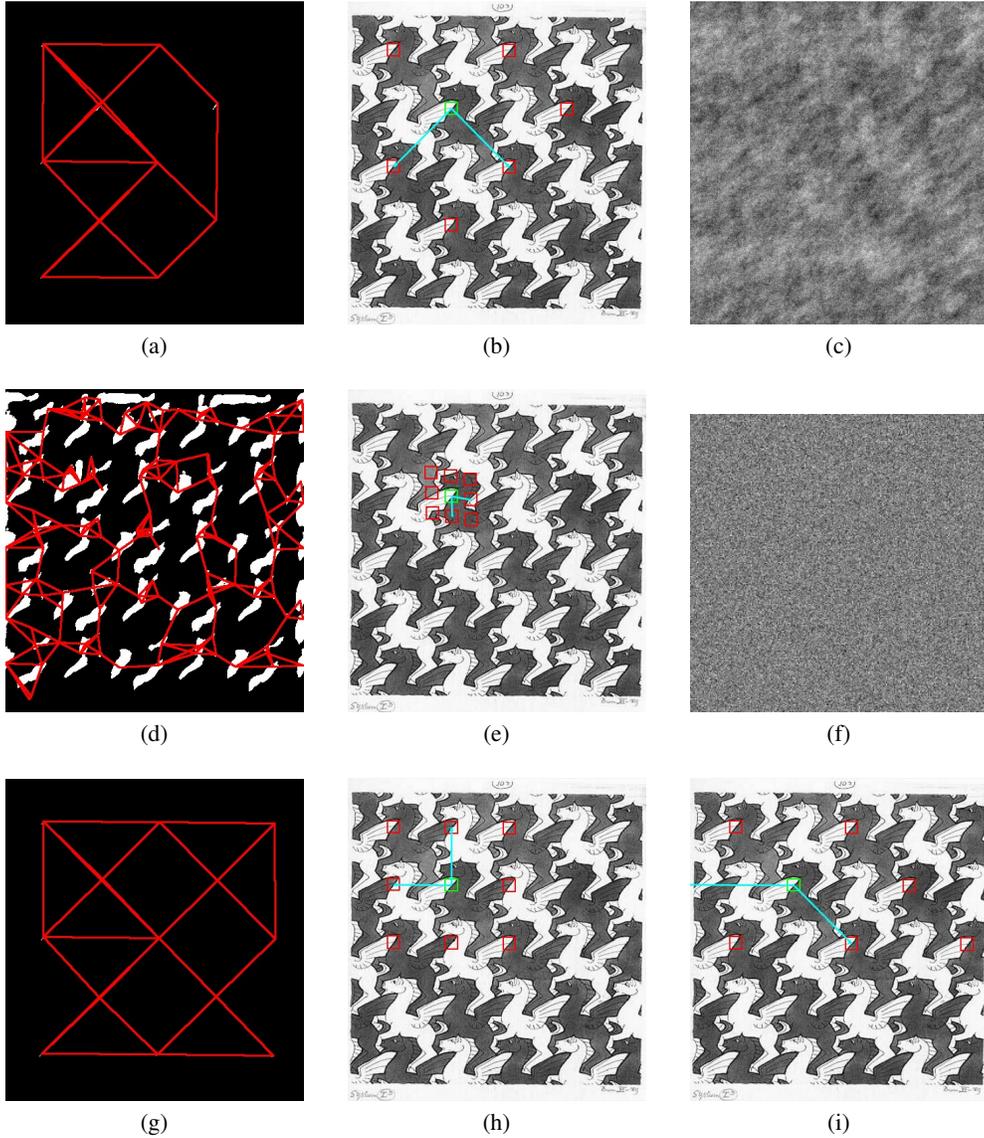


Figure 11: **Choice of the microtexture model.** In this experiment we discuss the choice of the *a contrario* background microtexture model. In the left column we display the graph obtained after the detection step. In the middle column we superpose the proposed lattice on the original image. The original 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 (7) and NFA_{\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), NFA_{\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 NFA_{\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 3 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).

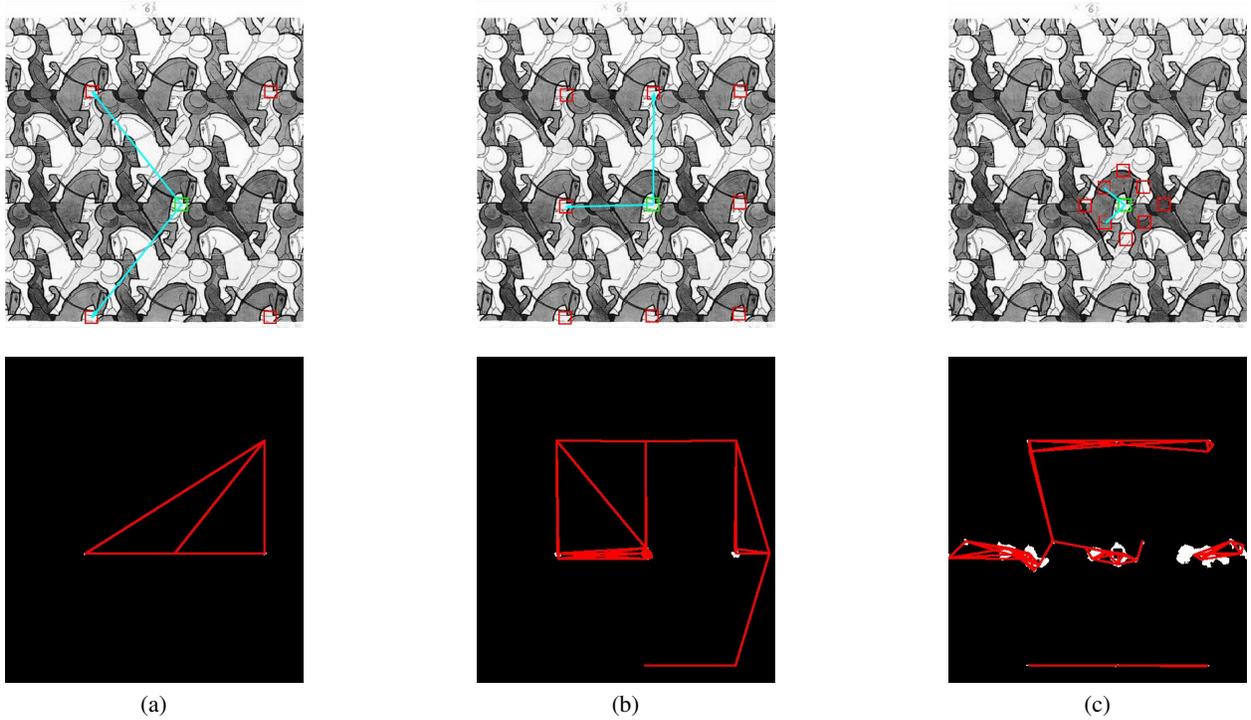


Figure 12: **Choice of Number of False Alarms.** In this experiment we discuss the choice of the NFA_{\max} parameter in the *a contrario* framework in the case where the underlying microtexture model is given by (7). Each column corresponds to a pair of images: the returned lattice and its associated underlying graph. In (a), NFA_{\max} is set to 1. Detections are correct, there are not enough points to precisely retrieve the perceptual lattice. In (b), NFA_{\max} is set to 10. The estimated lattice is correct. In (c), NFA_{\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.

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 camera position, see Figure 18.

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 Gaussian model with function f then $h * U$ is a sample from a Gaussian model with function $h * f$. Thus all the properties derived earlier remain valid with this linear operation. In Figure 17, we set h to be a Laplacian operator². 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.

²We use a discrete Laplacian operator Δ such that for any $\mathbf{x} = (x_1, x_2)$, we get that $\Delta(u)(x_1, x_2) = (u(x_1 + 1, x_2) + u(x_1 - 1, x_2) + u(x_1, x_2 + 1) + u(x_1, x_2 - 1) - 4u(\mathbf{x})) / 4$, where boundaries are handled periodically.

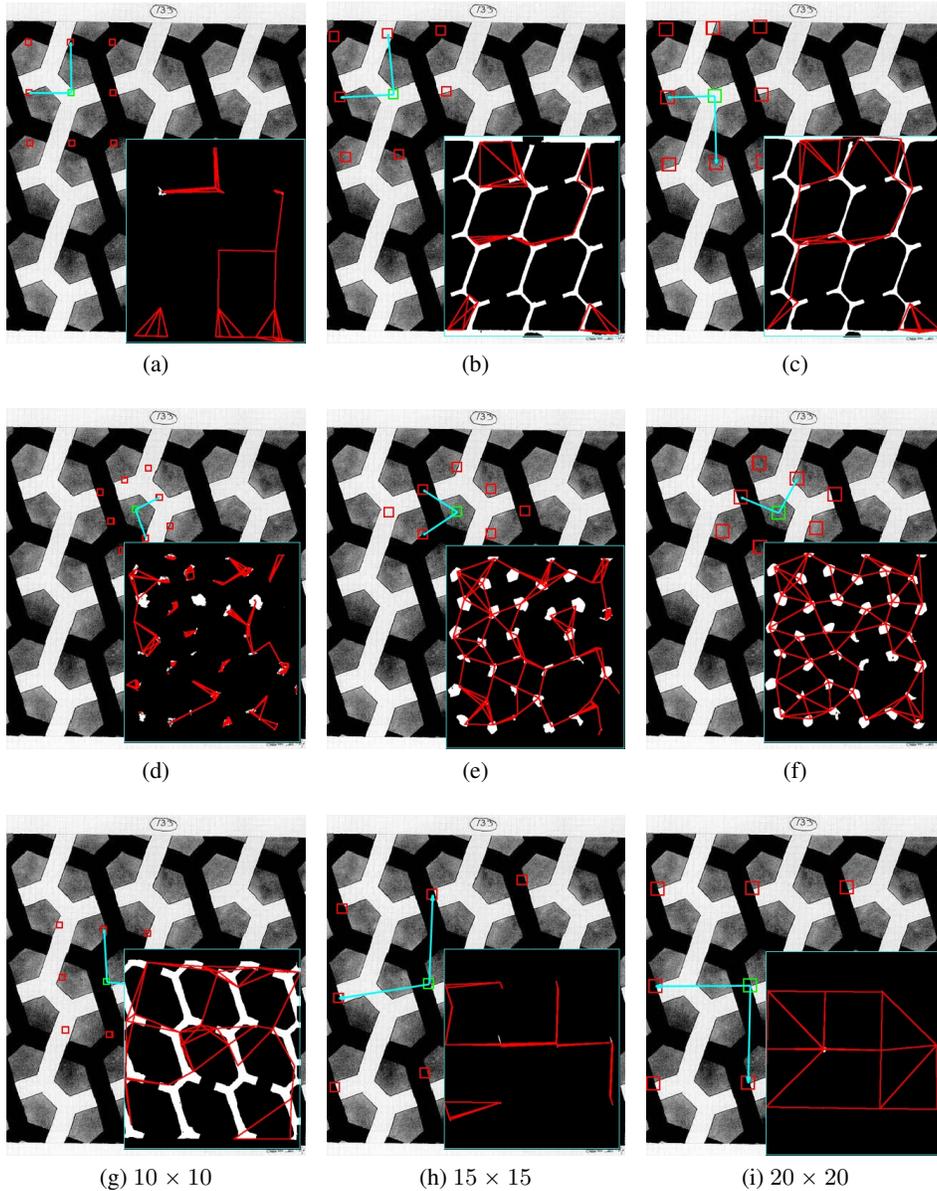


Figure 13: **Influence of patch size and patch position.** For each experiment NFA_{\max} is set to 10^4 , *i.e.* 4 % of the pixels. In most cases lower NFA_{\max} could be used but setting a high NFA_{\max} ensures that we always get detections even if the patch only contains microtexture information. Each row corresponds to a lattice proposal with same patch position but different patch sizes: 10×10 for the left column, 15×15 for the middle one and 20×20 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 10×10 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.

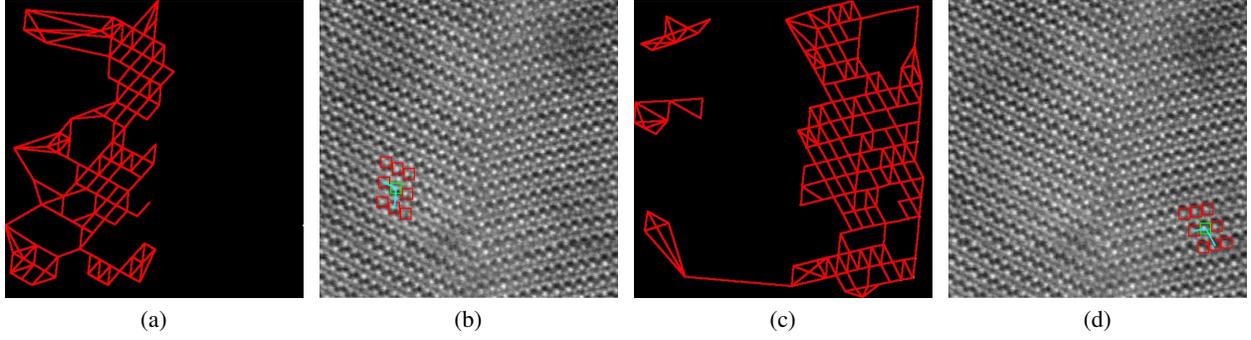


Figure 14: **Lattice extraction.** In this experiment we consider a crystallographic image (an orthorhombic NiZr alloy) and set NFA_{\max} to 10^2 . Two lattices are present in this image and they are correctly identified in (b) and (d). Note that in (a), respectively in (c), mostly points in the left, 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.

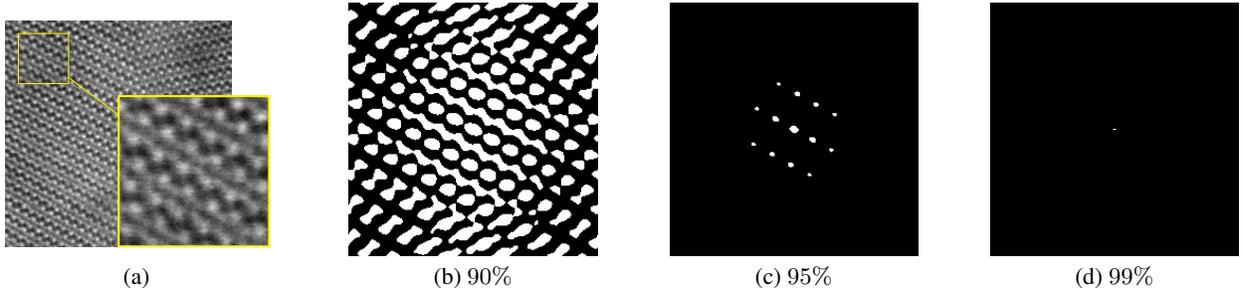


Figure 15: **Comparison with Fourier based methods.** Since the original image can be segmented in two highly periodic components, Fourier methods might be well-adapted to the lattice extraction task. In (a) we present a sub-image of the original alloy. We compute the autocorrelation of this sub-image and threshold it. This operation gives us a detection map, like Algorithm 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).

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 Ω , let $\omega \subset \Omega$ be a patch domain and a be as in Proposition 1 with NFA_{\max} set by the user.

Definition 6 (Periodicity criterion) Let $\{t \in \Omega, AS(u, t, \omega) \leq a(t)\}$ be the set of detected offsets and $N_{\mathcal{C}}$ its number of connected components as defined in Section 5.2. Let also $(\hat{B}, \hat{M}, \hat{\sigma})$ be the estimated parameters using Algorithm 3. We define the following periodicity criterion c_{per} as

$$c_{per}(u) = \frac{\pi \hat{\sigma}^2}{N_{\mathcal{C}} |\det(\hat{b}_1, \hat{b}_2)|}, \quad (15)$$

where $\hat{B} = (\hat{b}_1, \hat{b}_2)$.

The criterion c_{per} simply computes the ratio between the error area of Algorithm 3, *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.

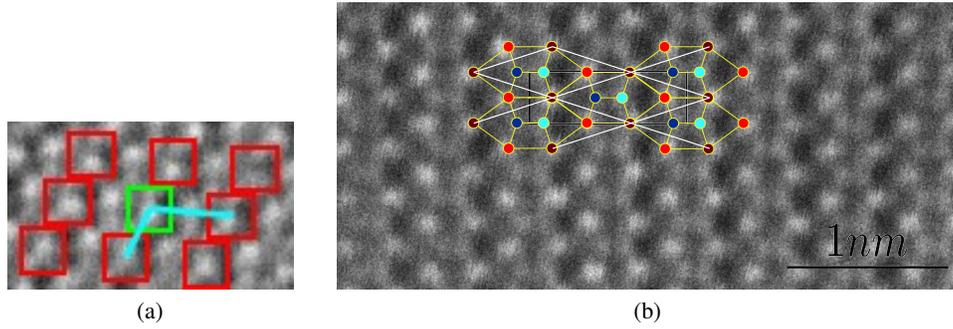


Figure 16: **Agreement with crystallography models.** In (a) we perform a zoom on of the lattice identified in Figure 14 and compare it to the one identified by crystallographists in (b). (a) is a zoomed rotated version of a crystalline structure similar to (b). The output lattice in (a) is the same 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.

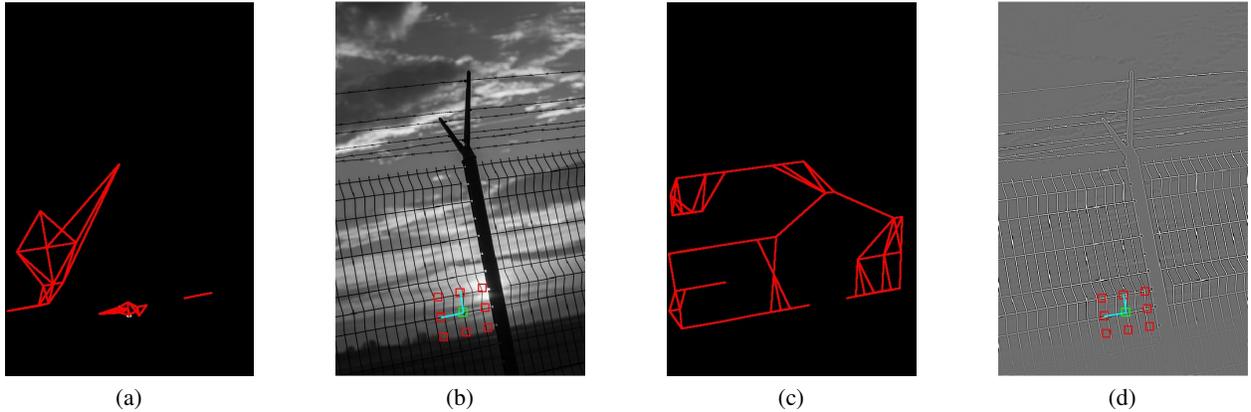


Figure 17: **Preprocessing and filtering.** In (a) and (c) we display the graphs obtained with Algorithm 1 applied on images (b) and (d). In (b) and (d) the original image is superposed with the estimated lattice (vectors in cyan and proposed patches in red). In (a) and (b), NFA_{\max} was set to 10^5 which corresponds to 35 % of detection in the associated *a contrario* model. Lower NFA_{\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 NFA_{\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.

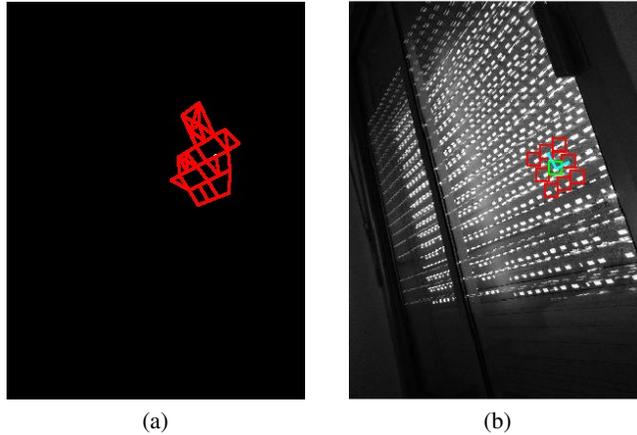


Figure 18: **Homography and locality.** In this experiment NFA_{\max} was set to 10^3 . Note that the detected graph is localized around the original patch in (a). In (b) we superpose the proposed lattice onto the original image. The lattice proposal is valid in a small neighborhood around the original patch. However it is not valid for the whole image.

In the experiment presented in Figure 19 we sort 25 texture images based on the c_{per} criterion. Images are of size 256×256 . 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 20×20 . For each set of parameters we find a lattice using Algorithm 1 and Algorithm 3 with parameters $NFA_{\max} = 1$, $\delta_M = 10$, $\delta_B = 10^{-2}$ and $N_{it} = 10$. 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 ℓ^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 [41] in order to estimate mean and covariance matrices with probabilistic guarantees. The generality of our model allows for several extensions for non-Gaussian noises [16] or to take into account the geometry of the patch space [34, 63].

Turning to periodicity detection we propose a novel graphical model using the output of Algorithm 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 finite-time convergence of Algorithm 3. 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 NFA_{\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 3. Since we set our background models to be Gaussian random fields and remarking that these are good microtexture approximations we wish to explore the possibility to embed our *a contrario*

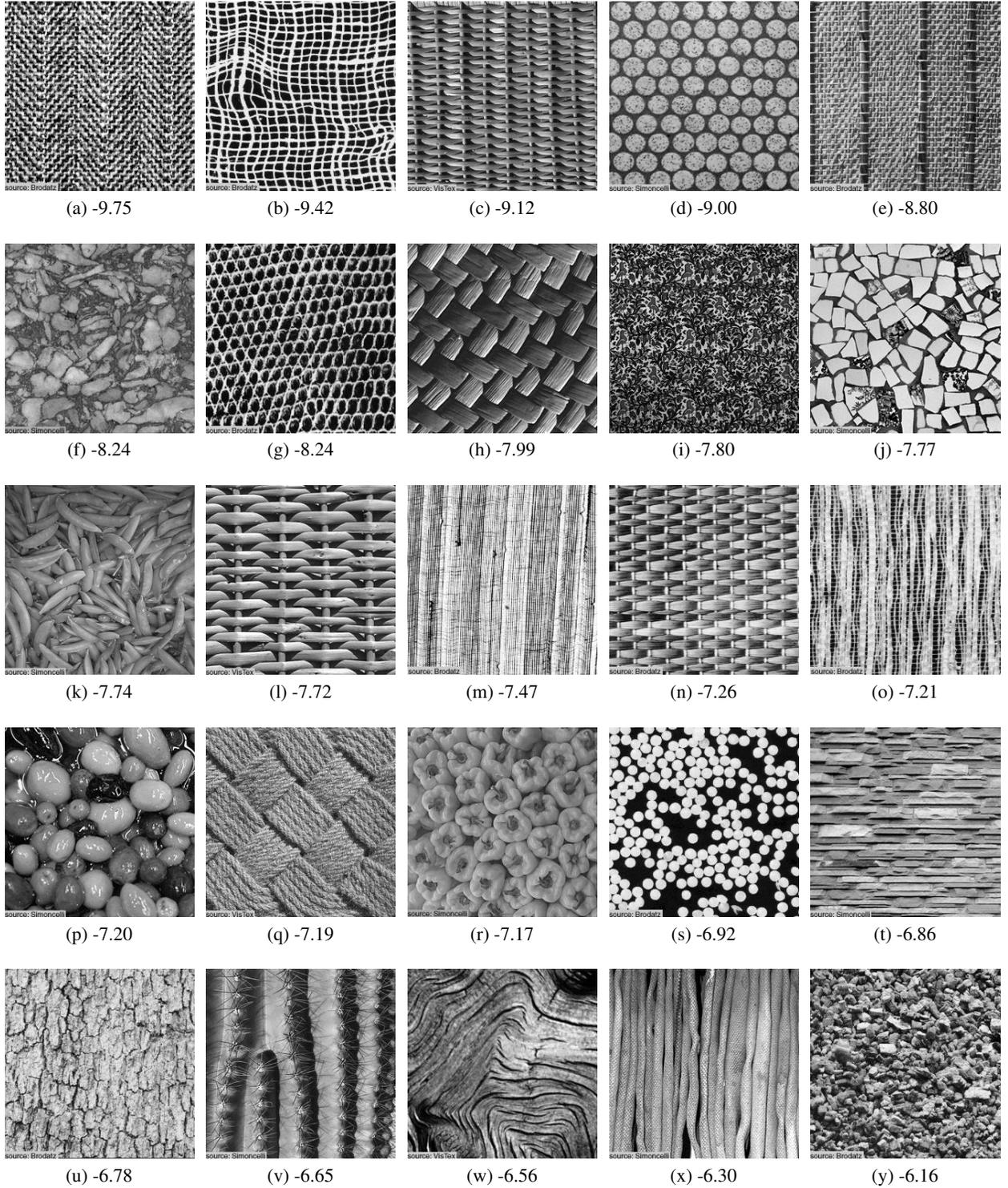


Figure 19: **Texture ranking.** The c_{per} criterion, defined in (15), is computed for each setting. We associate to each image the median of the 150 criterion values and sort the images accordingly. (a) corresponds to the lowest criterion, *i.e.* the most periodic image according to c_{per} criterion. (y) corresponds to the largest criterion, *i.e.* the least periodic image according to c_{per} . Under each image we give the logarithm of the median c_{per} values.

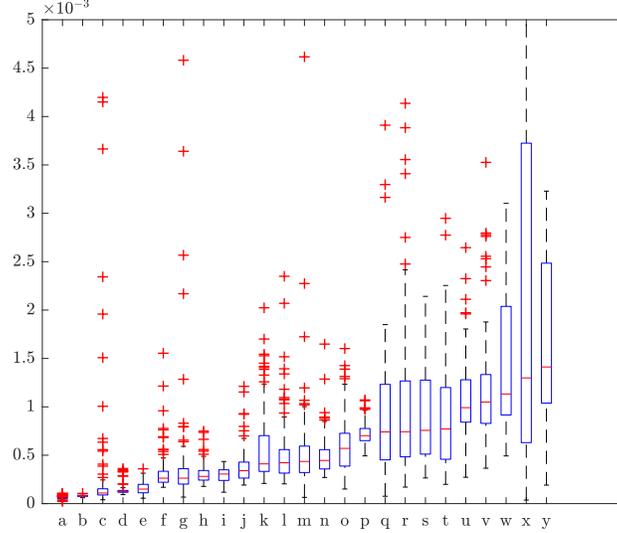


Figure 20: **Boxplot for c_{per} values.** In this figure we present a boxplot of the c_{per} values, defined in (15), used to rank textures images in Figure 19. We recall that we use 150 random patch positions in order to compute the c_{per} values. Letters on the x -axis correspond to the textures in Figure 19. For each texture we present its median c_{per} value. The lower, respectively upper, limit of the blue box corresponds to 25%, respectively 75% of the computed c_{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 5×10^{-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 20×20 patch. In order to perform a more robust ranking a multiscale approach should be preferred.

framework in texture analysis and texture synthesis algorithms. For instance an *a contrario* methodology could be incorporated in the algorithm proposed by Raad et al. in [56]. Another potential direction is to look at the behavior of the introduced dissimilarity functions for more general random fields in order to handle more complex and structured situations such as parametric texture synthesis.

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.

A Eigenvalues

Proof: [Proof of Proposition 5] We fix $\mathbf{t} \neq 0$ with $\|\mathbf{t}\|_\infty < p$ and denote $C = C_{\mathbf{t}}$. Without loss of generality we consider that $t_x > 0$ and $t_y > 0$. We consider X an eigenvector of C . Let $\Omega_0 = (\Omega - \mathbf{t}) \cap \Omega^c$ and the function $J : \Omega_0 \rightarrow \llbracket 2, +\infty \llbracket$ such that for any $\mathbf{x}_0 \in \Omega_0$

$$J(\mathbf{x}_0) = \operatorname{argmin} \{k \in \mathbb{N} \setminus \{0\}, \mathbf{x}_0 + k\mathbf{t} \notin \Omega\}.$$

It is clear that $I = \{(k, m), k \in \llbracket 1, m-1 \rrbracket, m \in J(\Omega_0)\}$ is in bijection with Ω . Let $\mathbf{x}_0 \in \Omega_0$, $m = J(\mathbf{x}_0)$ and $k \in \llbracket 1, m-1 \rrbracket$. We define $X_{\mathbf{x}_0, k}$ over \mathbb{Z}^2 such that

$$X_{\mathbf{x}_0, k}(\mathbf{x}_0 + \ell\mathbf{t}) = \sin\left(\frac{\ell k \pi}{m}\right) \text{ for } \ell \in \llbracket 1, m-1 \rrbracket, \quad 0 \text{ otherwise.}$$

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

$$X_{\mathbf{x}_0, k}(\mathbf{x} + \mathbf{t}) - 2\cos\left(\frac{k\pi}{m}\right) X_{\mathbf{x}_0, k}(\mathbf{x}) + X_{\mathbf{x}_0, k}(\mathbf{x} - \mathbf{t}) = 0.$$

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

$$2X_{\mathbf{x}_0,k}(\mathbf{x}) - X_{\mathbf{x}_0,k}(\mathbf{x} + \mathbf{t}) - X_{\mathbf{x}_0,k}(\mathbf{x} - \mathbf{t}) = \left[2 - 2 \cos \left(\frac{k\pi}{m} \right) \right] X_{\mathbf{x}_0,k}(\mathbf{x}) = 4 \sin^2 \left(\frac{k\pi}{m} \right) X_{\mathbf{x}_0,k}(\mathbf{x}).$$

Thus the one-dimensional vector (given by the raster-scan order on the x -axis) of the restriction of $X_{\mathbf{x}_0,k}$ is an eigenvector of C associated with eigenvalue $4 \sin^2 \left(\frac{k\pi}{m} \right)$.

Using that I is in bijection with Ω we get that the number of vectors $(X_{\mathbf{x}_0,k})$ is $|\Omega|$. We show that this family of vectors is linearly-independent. Let $\Lambda_{\mathbf{x}_0,k} \in \mathbb{R}$ such that

$$\sum_{\mathbf{x}_0 \in \Omega_0} \sum_{k=1}^{J(\mathbf{x}_0)-1} \Lambda_{\mathbf{x}_0,k} X_{\mathbf{x}_0,k} = 0.$$

Since $X_{\mathbf{x}_0,k}$ and $X_{\mathbf{y}_0,k'}$ have different support if and only if $\mathbf{x}_0 \neq \mathbf{y}_0$ we get that for any $\mathbf{x}_0 \in \Omega_0$, $\sum_{k=1}^{J(\mathbf{x}_0)-1} \Lambda_{\mathbf{x}_0,k} X_{\mathbf{x}_0,k} = 0$. This gives that $(\Lambda_{\mathbf{x}_0,k})_{k \in \llbracket 1, J(\mathbf{x}_0)-1 \rrbracket}$ is in the kernel of the matrix $(\sin(\ell k \pi / (J(\mathbf{x}_0) - 1)))_{1 \leq j, \ell \leq J(\mathbf{x}_0)-1}$. Since the sinus discrete transform is invertible we obtain that for any $\mathbf{x}_0 \in \Omega_0$ and $k \in \llbracket 1, J(\mathbf{x}_0) - 1 \rrbracket$, $\Lambda_{\mathbf{x}_0,k} = 0$. Thus the family $X_{\mathbf{x}_0,k}$ is a basis of eigenvectors.

We aim at computing the cardinality of $K_{k,m} = \{X_{\mathbf{x}_0,k}, J(\mathbf{x}_0) = m\}$. 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

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

Note that for all $\mathbf{x}_0 \in \Omega_0$ we have that $\mathbf{x}_0 + (q+1)\mathbf{t} \notin \Omega$, with $q = \lceil \frac{p}{|t_x| \vee |t_y|} \rceil$. 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 $\mathbf{x}_0 \in \Omega_x$ we have

$$\mathbf{x}_0 = (i_0, j_0) \in K_{k,m} \Leftrightarrow \begin{cases} i_0 + mt_x \geq p \\ \text{or} \\ j_0 + mt_y \geq p \end{cases} \quad \text{and} \quad \begin{cases} i_0 + (m-1)t_x \leq p-1 \\ \text{and} \\ j_0 + (m-1)t_y \leq p-1 \end{cases}.$$

Since $\mathbf{x}_0 \in \Omega_x$ we have $i_0 + mt_x \leq p-1$, hence

$$\mathbf{x}_0 = (i_0, j_0) \in K_{k,m} \Leftrightarrow j_0 + mt_y \geq p \text{ and } j_0 + (m-1)t_y \leq p-1.$$

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

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 $\mathbf{x}_0 \in \Omega_{x,y}$ then $\mathbf{x}_0 = -\mathbf{t} + (x, y)$ with $x \in \llbracket 0, t_x - 1 \rrbracket$ and $y \in \llbracket 0, t_y - 1 \rrbracket$. We obtain the following equivalence

$$\mathbf{x}_0 \in J^{-1}(q+1) \Leftrightarrow \begin{cases} -t_x + x + (q+1)t_x \geq p \\ \text{or} \\ -t_y + y + (q+1)t_y \geq p \end{cases} \quad \text{and} \quad \begin{cases} -t_x + x + qt_x \leq p-1 \\ \text{and} \\ -t_y + y + qt_y \leq p-1 \end{cases}.$$

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

$$\mathbf{x}_0 \in J^{-1}(q+1) \Leftrightarrow x \leq p-1 - (q-1)t_x \text{ and } y \leq p-1 - (q-1)t_y.$$

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

B Update rules

We derive the proof of Proposition 7.

Proof: Computing the minimum of $q(B, M|E)$ for fixed $B \in \mathbb{R}^4$, respectively fixed $M \in \mathbb{R}^{2|E|}$, gives the update rule for M , respectively for B . We obtain that

$$\begin{aligned}
q(B, M|E) &= \sum_{\mathbf{e} \in E} m_{\mathbf{e}}^2 \|b_1\|^2 + \sum_{\mathbf{e} \in Eb} n_{\mathbf{e}}^2 \|b_2\|^2 + 2 \sum_{\mathbf{e} \in E} m_{\mathbf{e}} n_{\mathbf{e}} \langle b_1, b_2 \rangle \\
&\quad - 2 \sum_{\mathbf{e} \in E} m_{\mathbf{e}} \langle b_1, \mathbf{e} \rangle - 2 \sum_{\mathbf{e} \in E} n_{\mathbf{e}} \langle b_2, \mathbf{e} \rangle + r(B, M) \\
&= B^T (\Lambda_M \otimes \text{Id}_2) B - 2 \langle B, E_M \rangle + \alpha(M) \\
&= \|(\Lambda_M \otimes \text{Id}_2)^{\frac{1}{2}} B - (\Lambda_M \otimes \text{Id}_2)^{-\frac{1}{2}} E_M\|^2 + \alpha(M) \\
&= \|(\Lambda_M \otimes \text{Id}_2)^{\frac{1}{2}} \left(B - (\Lambda_M \otimes \text{Id}_2)^{-1} E_M \right)\|^2 + \alpha(M),
\end{aligned}$$

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

References

- [1] Andrés Almansa, Agnès Desolneux, and Sébastien Vamech. Vanishing point detection without any A priori information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(4):502–507, 2003.
- [2] Suyash P. Awate and Ross T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):364–376, 2006.
- [3] Dean A. Bodenham and Niall M. Adams. A comparison of efficient approximations for a weighted sum of chi-squared random variables. *Stat. Comput.*, 26(4):917–928, 2016.
- [4] J. Bruna and S. Mallat. Multiscale Sparse Microcanonical Models. *ArXiv e-prints*, January 2018.
- [5] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 60–65, 2005.
- [6] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-Local Means Denoising. *Image Processing On Line*, 1:208–212, 2011.
- [7] Frédéric Cao. Application of the Gestalt principles to the detection of good continuations and corners in image level lines. *Comput. Vis. Sci.*, 7(1):3–13, 2004.
- [8] Richard W. Connors and Charles A. Harlow. Toward a structural textural analyzer based on statistical methods. *Computer Graphics and Image Processing*, 12(3):224–256, 1980.
- [9] David Coupier, Agnès Desolneux, and Bernard Ycart. Image denoising by statistical area thresholding. *Journal of Mathematical Imaging and Vision*, 22(2-3):183–197, 2005.
- [10] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Processing*, 13(9):1200–1212, 2004.
- [11] George R. Cross and Anil K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5(1):25–39, 1983.
- [12] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, 2007.
- [13] John G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, Jul 1985.
- [14] Axel Davy, Thibaud Ehret, Jean-Michel Morel, and Mauricio Delbracio. Reducing anomaly detection in images to detection in noise. In *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*, pages 1058–1062, 2018.
- [15] Valentin De Bortoli, Agnes Desolneux, Bruno Galerne, and Arthur Leclaire. Redundancy in Gaussian random fields. *Submitted*, Nov 2018.
- [16] Charles-Alban Deledalle, Loïc Denis, and Florence Tupin. Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. Image Processing*, 18(12):2661–2672, 2009.
- [17] Charles-Alban Deledalle, Loïc Denis, and Florence Tupin. How to compare noisy patches? Patch similarity beyond Gaussian noise. *Int. J. Comput. Vis.*, 99(1):86–102, 2012.

- [18] Julie Delon and Agnès Desolneux. A patch-based approach for removing impulse or mixed Gaussian-impulse noise. *SIAM J. Imaging Sciences*, 6(2):1140–1174, 2013.
- [19] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.
- [20] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Edge Detection by Helmholtz Principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.
- [21] Agnes Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt theory to image analysis: a probabilistic approach*, volume 34. Springer Science & Business Media, 2007.
- [22] Vincent Duval, Jean-François Aujol, and Yann Gousseau. On the parameter choice for the Non-Local Means. 37 pages., March 2010.
- [23] Vincent Duval, JFean-rançois Aujol, and Yann Gousseau. A bias-variance approach for the nonlocal means. *SIAM J. Imaging Sciences*, 4(2):760–788, 2011.
- [24] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 341–346, 2001.
- [25] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999.
- [26] Ronan Fablet, J.-M Augustin, and Alexandru Isar. Speckle denoising using a variational multi-wavelet approach. In *Oceans 2005 - Europe*, volume 1, pages 539 – 544 Vol. 1, 07 2005.
- [27] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random phase textures: Theory and synthesis. *IEEE Trans. Image Processing*, 20(1):257–267, 2011.
- [28] Bruno Galerne, Arthur Leclaire, and Julien Rabin. A Texture Synthesis Model Based on Semi-discrete Optimal Transport in Patch Space. working paper or preprint, March 2018.
- [29] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 262–270, 2015.
- [30] Bart Goossens, Hiep Luong, Aleksandra Pizurica, and Wilfried Philips. An improved non-local denoising algorithm. In Jaakko Astola, Karen Egiazarian, and Vladimir Katkovnik, editors, *Local and Non-Local Approximation in Image Processing, International Workshop, Proceedings*, pages 143–156, 2008.
- [31] Sven Grewenig, Sebastian Zimmer, and Joachim Weickert. Rotationally invariant similarity measures for nonlocal image denoising. *J. Visual Communication and Image Representation*, 22(2):117–130, 2011.
- [32] Robert M. Haralick, K. Sam Shanmugam, and Its’hak Dinstein. Textural features for image classification. *IEEE Trans. Systems, Man, and Cybernetics*, 3(6):610–621, 1973.
- [33] Kaiming He and Jian Sun. Image completion approaches using the statistics of similar patches. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(12):2423–2435, 2014.
- [34] A. Houdard, C. Bouveyron, and J. Delon. High-dimensional mixture models for unsupervised image denoising (HDMI). *SIAM Journal on Imaging Sciences*, 11(4):2815–2846, 2018.
- [35] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat’s striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.
- [36] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48:419–426, 1961.
- [37] Charles Kervrann and Jérôme Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.
- [38] Vivek Kwatra, Arno Schödl, Irfan A. Essa, Greg Turk, and Aaron F. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.
- [39] M. Lebrun, M. Colom, A. Buades, and J. M. Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.
- [40] Marc Lebrun. An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line*, 2:175–213, 2012.
- [41] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A Nonlocal Bayesian Image Denoising Algorithm. *SIAM J. Imaging Sciences*, 6(3):1665–1688, 2013.
- [42] Arthur Leclaire. *Champs à phase aléatoire et champs gaussiens pour la mesure de netteté d’images et la synthèse rapide de textures*. PhD thesis, Université Paris Descartes, 2015.

- [43] Hsin-Chih Lin, Ling-Ling Wang, and Shi-Nine Yang. Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern Recognition Letters*, 18(5):433–443, 1997.
- [44] Huan Liu, Yongqiang Tang, and Hao Helen Zhang. A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central normal variables. *Computational Statistics & Data Analysis*, 53(4):853–856, 2009.
- [45] Yanxi Liu, Robert T. Collins, and Yanghai Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(3):354–371, 2003.
- [46] David G. Lowe and Thomas O. Binford. Perceptual organization as a basis for visual recognition. In *Proceedings of the National Conference on Artificial Intelligence*, pages 255–260, 1983.
- [47] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning FRAME models using CNN filters. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1902–1910, 2016.
- [48] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI*, pages 674–679, 1981.
- [49] Takashi Matsuyama, Shu-Ichi Miura, and Makoto Nagao. Structural analysis of natural textures by fourier transformation. *Computer Vision, Graphics, and Image Processing*, 24(3):347–362, 1983.
- [50] Niklas Mevenkamp and Benjamin Berkels. Unsupervised and accurate extraction of primitive unit cells from crystal images. In *Pattern Recognition - 37th German Conference, GCPR 2015*, pages 105–116, 2015.
- [51] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2001.
- [52] Gyuhan Oh, Seungyong Lee, and Sung Yong Shin. Fast determination of textural periodicity using distance matching function. *Pattern Recognition Letters*, 20(2):191–197, 1999.
- [53] Minwoo Park, Kyle Brocklehurst, Robert T. Collins, and Yanxi Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1804–1816, 2009.
- [54] Nikolay N. Ponomarenko, Vladimir V. Lukin, Mikhail Zriakhov, Arto Kaarna, and Jaakko Astola. An automatic approach to lossy compression of AVIRIS images. In *IEEE International Geoscience & Remote Sensing Symposium, IGARSS 2007, July 23-28, 2007, Barcelona, Spain, Proceedings*, pages 472–475, 2007.
- [55] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.
- [56] Lara Raad, Agnès Desolneux, and Jean-Michel Morel. A conditional multiscale locally gaussian texture synthesis algorithm. *Journal of Mathematical Imaging and Vision*, 56(2):260–279, 2016.
- [57] Joseph Salmon. On two parameters for denoising with non-local means. *IEEE Signal Process. Lett.*, 17(3):269–272, 2010.
- [58] Xiahan Sang and James M. LeBeau. Revolving scanning transmission electron microscopy: Correcting sample drift distortion without prior knowledge. *Ultramicroscopy*, 138:28 – 35, 2014.
- [59] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1233–1240, 2013.
- [60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.
- [61] Frédéric Sur. An a-contrario approach to quasi-periodic noise removal. In *2015 IEEE International Conference on Image Processing*, pages 3841–3845, 2015.
- [62] Jarke J. van Wijk. Spot noise texture synthesis for data visualization. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 309–318, 1991.
- [63] Yi-Qing Wang and Jean-Michel Morel. SURE guided gaussian mixture image denoising. *SIAM J. Imaging Sciences*, 6(2):999–1034, 2013.
- [64] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems Computers*, volume 2, pages 1398–1402 Vol.2, Nov 2003.
- [65] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004.

- [66] Andrew T. A. Wood. An f approximation to the distribution of a linear combination of chi-squared variables. *Communications in Statistics - Simulation and Computation*, 18(4):1439–1456, 1989.
- [67] Yue Wu, Brian Tracey, Premkumar Natarajan, and Joseph P. Noonan. Probabilistic non-local means. *IEEE Signal Process. Lett.*, 20(8):763–766, 2013.
- [68] Gui-Song Xia, Sira Ferradans, Gabriel Peyré, and Jean-François Aujol. Synthesizing and mixing stationary gaussian texture models. *SIAM J. Imaging Sciences*, 7(1):476–508, 2014.
- [69] Song Chun Zhu. Embedding Gestalt laws in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1170–1187, 1999.
- [70] Song Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.
- [71] Steven W. Zucker and Demetri Terzopoulos. Finding structure in co-occurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12(3):286 – 308, 1980.