PATCH REDUNDANCY IN IMAGES: A STATISTICAL TESTING FRAMEWORK AND SOME APPLICATIONS

Valentin de Bortoli, Agnès Desolneux, Bruno Galerne, Arthur Leclaire

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.

November 21, 2018

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.

( )

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

1 Introduction

In many image processing applications using local information coupled with long-range correlation 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. For example, patch-based inpainting techniques, such as [1, 2], assign patches of the known region to patches of the 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 [2] the authors replace the search on the whole image, which could be accelerated with a PatchMatch algorithm [3], by a search among the most redundant offsets in the known region. This allows the authors of [2] to retrieve long-range spatial structure in the unknown zone. Another famous application of the exploitation of spatial redundancy can be found in denoising with the seminal work (Non-Local means) of Buades et al. [4], 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. Indeed, long-range correlation is lost in microtexture models where the only structural information is given by the first-order and second-order statistics of the exemplar image. These microtexture models can be described by Gaussian random fields [5, 6, 7, 8]. In this case the loss of long-range spatial structures is explained by the randomization of the Fourier phase. Parametric models using features such as wavelet transform coefficients [9], scattering transform coefficients [10] or convolutional neural network outputs [11] have been proposed in order to retrieve the lost spatial information. On the other hand, non-parametric patch-based algorithms such as [12, 13, 14] propose to use most similar patches in order to fill the new texture images, similarly to inpainting techniques.

All these techniques lift images in higher dimensional spaces, making 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 human vision [15]. These last years have seen the rise of neural networks in which the filter basis is no longer given as an input but learned through a data-driven optimization procedure [16]. On the other hand, patch-based methods rely on the assumption that image processing tasks are simplified when conducted in a higher dimensional patch space. Indeed, patches are local measurements which contain structural information and their spatial arrangement provides knowledge about large-scale arrangement of the image. In their seminal paper, Efros and Leung [12] show that patch information was sufficient to synthesize a wide variety of textures. Since then, patch information was successfully used in many image processing tasks including inpainting [1], denoising [4] and texture synthesis [13, 17, 14, 18].

Every analysis performed in a lifted space, built via feature extraction or patch extraction, relies on the comparison of points in this space. Thus different metrics or similarity scores yield different resemblance notions. In this paper, we focus only on patch-based methods and propose similarity functions to evaluate if two points in the lifted space, *i.e.* two patches, are similar or not. In patch-based lifted spaces, we aim at finding similarity functions such that two patches are said to be close if they are perceptually close. Usually the squared Euclidean norm is preferred, however this measurement is a poor choice if the lifted space is high dimensional, due to the well-known curse of dimensionality. Measures relying on order two statistics were proposed in order to correct these drawbacks [19, 20] and give more perceptually satisfying measurements. In this paper, we follow the work of [21] in which the authors propose new similarity functions, defined by a few properties, for noisy images. Since noise in images can be modeled by a random field, similarity functions can be designed as statistical measurements.

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 [5, 6, 7, 8] or spatial Markov random fields [22], 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 [23, 24] or some stochastic optimization procedure [25]. This leads 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 [26, 27, 28, 29, 30] and aims at identifying structure events in images. This statistical model takes its roots in the fundamental work of the Gestalt theory [31]. One of its principle, the non-accidentalness principle [32] or Helmholtz principle [33, 28], states that no structure is perceived in a Gaussian white noise. This constitutes a first example of the *a contrario* methodology which aims at identifying quantitative conditions under which a background model, a Gaussian white noise model in the Helmholtz principle, can be rejected. 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 local auto-similarity 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 [4] (NL-means) by inserting a threshold in the selection of similar patches. Using an *a contrario* model we are able to give probabilistic controls 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 image analysis in early mathematical vision [34]. Most proposed methods to analyze periodicity rely on global measurements such as the modulus of the Fourier transform [35] or the autocorrelation [36]. These global techniques are widely used in crystallography where lattice properties, such as the angle between basis vectors, are fundamental [37, 38]. 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 similarity and the inertia measurement in co-occurence matrices [34]. 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 [39], 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 criterions in order to differentiate textures. In [40], the authors use a classic 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. Section 2 is dedicated to the introduction of similarity functions and *a contrario* models. In Section 2.1 we introduce similarity functions on patches, as local measurements on images. An *a contrario* framework for local similarity detection is then proposed in Section 2.2. In the *a contrario* framework, a background model, corresponding to the null hypothesis, is required. In Section 3.1 we justify our choice of Gaussian random fields as noise models. The consequence of choosing Gaussian models as reference models are investigated and a redundancy detection algorithm is proposed in Section 3.2. 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 similarity 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 6, we introduce a criterion for measuring texture periodicity. We conclude our study and discuss future work in Section 7.

2 Similarity functions and a contrario framework

2.1 Similarity functions

In order to evaluate redundancy in images we first need to derive a criterion for comparing images. We propose to use similarity functions which take two images, or sub-images, as inputs. Before discussing the choice of function, or patch domain, we must precise which images are going to be used to conduct the redundancy detection.

Indeed, when comparing sub-images, two cases can occur. We can compare a patch with another patch extracted from the same image. We call this situation *internal matching*. Applications can be found in denoising [4] or inpainting [1] where the information of the image itself is used to perform the image processing task. On the other hand, we can compare a patch with a patch extracted from another image. We call this situation *template matching*. An application of this case is presented in the non-parametric exemplar texture synthesis algorithm proposed by Efros and Leung [12].

The Euclidean distance is the usual way to measure similarity between patches [41] but many other measurements exist, corresponding to different structural properties, see Figure 1. We introduce p-norms and angle measurements similarity functions.

Definition 1 *Let $P, Q \in \mathbb{R}^\omega$ with ω a finite domain of $\mathbb{Z}^2$. When it is defined we introduce*

(a) the ℓ^p-similarity, $s_p(P,Q) = \|P-Q\|_p = \left(\sum_{\mathbf{x}\in\omega} |P(\mathbf{x})-Q(\mathbf{x})|^p\right)^{1/p}$, with $p \in (0,+\infty)$;

(b) the ℓ^∞-similarity, $s_\infty(P,Q) = \sup_\omega(|P-Q|)$;

(c) the p-th power of the ℓ^p-similarity, $s_{p,p}(P,Q) = s_p(P,Q)^p$, with $p \in (0,+\infty)$;

(d) the scalar product similarity, $s_{sc}(P,Q) = -\langle P,Q\rangle = \frac{1}{2}\left(s_{2,2}(P,Q) - \|P\|_2^2 - \|Q\|_2^2\right)$;

(e) the cosine similarity, $s_{\cos}(P,Q) = \frac{s_{sc}(P,Q)}{\|P\|_2\|Q\|_2}$, if $\|P\|_2\|Q\|_2 \neq 0$.

The locality of the measurements is ensured by the fact that these functions are defined on patches, *i.e.* sub-images defined on domain ω. Following conditions (1) and (3) in [21] we check that similarity functions (a), (c) and (e) satisfy the following properties

- (Symmetry) $s(P,Q) = s(Q,P)$;
- (Maximal self-similarity) $s(P,P) \leqslant s(P,Q)$;
- (Equal self-similarities) $s(P,P) = s(Q,Q)$.

Note that since s_{sc}, the scalar product similarity, is homogeneous in P, maximal self-similarity and equal self-similarity properties are not satisfied. In [21], the authors present many other similarity functions all relying on statistical

properties such as likelihood ratios, joint likelihood criteria and mutual information kernels. The latter measurement is defined by a cosine measurement in some feature space. In this paper we focus on similarity functions defined directly in the spatial domain. The limits of this restriction will be investigated in Section 5.3.3.

Definition 2 (Auto-similarity and template similarity) *Let u and v be two images defined over an image domain $\Omega = [\![0, M-1]\!] \times [\![0, N-1]\!] \subset \mathbb{Z}^2$, with $M, N \in \mathbb{N}$. 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 ω and offset $\mathbf{t} \in \mathbb{Z}^2$ by*

$$\mathcal{AS}_i(u, \mathbf{t}, \omega) = s_i\left(P_{\mathbf{t}+\omega}(u), P_\omega(u)\right), \tag{1}$$

where s_i corresponds to s_p with $p \in (0, +\infty]$, $s_{p,p}$ with $p \in (0, +\infty)$, s_{sc} or s_{cos}. In the same way we define the template similarity with patch ω and offset $\mathbf{t}$ by

$$\mathcal{TS}_i(u, v, \mathbf{t}, \omega) = s_i\left(P_{\mathbf{t}+\omega}(u), P_\omega(v)\right). \tag{2}$$

The auto-similarity computes the local resemblance 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 $\mathbf{t}$, whereas the template similarity uses an image v as input and computes local resemblances between u and v.

Suppose we evaluate the scalar product auto-similarity $\mathcal{AS}_{sc}(U, \mathbf{t}, \omega)$ with U a random field. Then the expectation of the auto-similarity function depends on second-order statistics of U. In the template case, the expectation of $\mathcal{AS}_{sc}(U, v, \mathbf{t}, \omega)$ depends on first-order statistics of U. This shows that auto-similarity and template similarity can exhibit very different statistical behaviors even for the same similarity functions.

It is well-known that, due to the curse of dimensionality, the ℓ^2 norm does not behave well in large-dimensional spaces and is a poor measure of structure. Thus, considering u and v two images, $s_2(u, v)$, the ℓ^2 template similarity on full images, does not yield interesting information about the perceptual differences between u and v. The template similarity $\mathcal{TS}_2(u, v, \mathbf{0}, \omega)$ avoids this effect by considering patches, thus reduncing the dimension of the data (if the cardinality of ω, denoted $|\omega|$, is small) and also allows for fast computation of similarity mappings. Even when considering small patches there exist other limitations such as noise and outliers sensitivity or illumination dependence, see Figure 1.

In the following section, we introduce an *a contrario* framework on similarity functions. This framework will allow us to derive an algorithm for detecting spatial redundancy in natural images.

2.2 A contrario framework

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 similarity 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 similarity 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 [31] 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 [33] or the non-accidentalness principle [32]. Therefore, in order to control the number of similarities identified in the background model, we study the probability density function of the auto-similarity and template similarity functions with input random image U over Ω. We will denote by $\mathbb{P}_0$ the probability distribution of U over $\mathbb{R}^\Omega$, the images over Ω. We will assume that $\mathbb{P}_0$ is a microtexture model, *i.e.* U does not present long-range correlation. Consequently, exhibiting repetitive structure at long-range is a rare event in a microtexture model. We define the following significant events which encode spatial redundancy:

- auto-similarity event: $\mathcal{AS}_i(u, \mathbf{t}, \omega) \leqslant a(\mathbf{t})$;
- template-similarity event: $\mathcal{TS}_i(u, v, \mathbf{t}, \omega) \leqslant a(\mathbf{t})$;

where a, the threshold function, is defined over the offsets ($\mathbf{t} \in \mathbb{Z}^2$) but might also depends on other parameters such as ω, $\mathbb{P}_0$ or v.

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 $\mathbb{P}_0$. In our model the significant event is patch

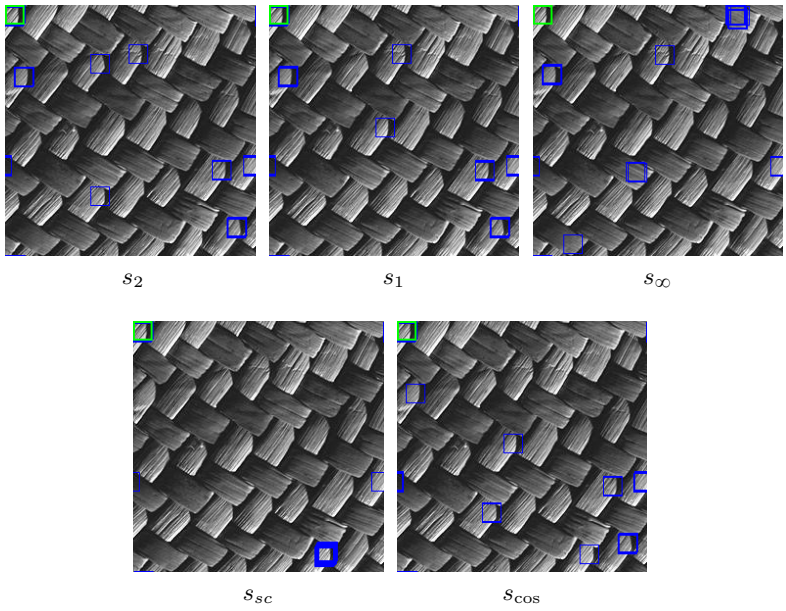


Figure 1: ***Structural properties of similarity functions.*** Sub-images show the twenty closest matches in the 20×20 patch space for the green upper-left patch in the original image for different similarity functions. All similarity functions correctly identify the structure of the patch, *i.e.* a large clear part with diagonal textures and a dark ray on the right side of the patch, except for s_∞ which is too sensitive to outliers. Indeed outliers have more importance for s_∞ than they have perceptually speaking. Similarities s_2 and s_1 have analogous behaviors and find correct regions. It might be noted that s_1 is more conservative as it identifies 7 main different patches and s_2 identifies eight. Similarity s_{sc} is too sensitive to contrast and, as it finds a correct patch, it gives too much importance to illumination. The behavior of $s_{\cos}$ is interesting as it avoids some of the illumination problems encountered with the scalar product. The identified regions were also found with s_1 and s_2, but with the addition of a new one.

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 definitions of the NFA in the spatial redundancy context.

Definition 3 (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*

$$\mathsf{AP}_i(\mathbf{t}, \omega, a) = \mathbb{P}_0 \left[\mathcal{AS}_i(U, \mathbf{t}, \omega) \leqslant a(\mathbf{t}) \right] .$$

In the same way we introduce the template similarity probability map TP *for $v \in \mathbb{R}^\Omega$ by*

$$\mathsf{TP}_i(v, \mathbf{t}, \omega, a) = \mathbb{P}_0 \left[\mathcal{TS}_i(U, v, \mathbf{t}, \omega) \leqslant a(\mathbf{t}) \right] .$$

We define the auto-similarity expected number of false alarms ANFA *and template similarity expected number of false alarms* TNFA *by*

$$\mathsf{ANFA}_i(\omega, a) = \sum_{\mathbf{t} \in \Omega} \mathsf{AP}_i(\mathbf{t}, \omega, a) \qquad \text{and} \quad \mathsf{TNFA}_i(v, \omega, a) = \sum_{\mathbf{t} \in \Omega} \mathsf{TP}_i(v, \mathbf{t}, \omega, a) . \tag{3}$$

Note that $\mathsf{AP}_i(\mathbf{t}, \omega, a)$ 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 $\mathcal{AS}_i(U, \mathbf{t}, \omega)$ under $\mathbb{P}_0$ evaluated at value α is given by $\mathsf{AP}_i(\mathbf{t}, \omega, \alpha)$. We denote by $q \mapsto \mathsf{AP}_i^{-1}(\mathbf{t}, \omega, q)$ the inverse cumulative distribution function, potentially defined by a generalized inverse ($\mathsf{AP}_i^{-1}(\mathbf{t}, \omega, q) = \inf_{\alpha \in \mathbb{R}} \mathsf{AP}_i(\mathbf{t}, \omega, \alpha) \geqslant q$), of the auto-similarity random variable for a fixed offset $\mathbf{t}$, with $q \in (0, 1)$ a quantile. The inverse cumulative distribution function of the template similarity function under $\mathbb{P}_0$ is defined in the same manner. We now have all the tools to control the number of detected offsets in the background model.

Definition 4 (Detected offset) *Let $u \in \mathbb{R}^{\Omega}$ and image, $\omega \subset \Omega$ a patch domain, and $a \in \mathbb{R}^{\Omega}$. An offset $\mathbf{t}$ is said to be detected, in the auto-similarity case, if $\mathcal{AS}_i(u, \mathbf{t}, \omega) \leqslant a(\mathbf{t})$. A similar definition is derived for template similarity functions given $v \in \mathbb{R}^{\Omega}$.*

Note that a detected offset in the background model $\mathbb{P}_0$ corresponds to a false alarm in the *a contrario* model. The next proposition defines a function a such that the expected number of false alarms, *i.e.* the expected number of detected offsets, is controlled in the null hypothesis case. In what follows we suppose that the cumulative distribution function of $\mathcal{AS}_i(U, \mathbf{t}, \omega)$ is invertible for every $\mathbf{t} \in \Omega$. This ensures that for any $\mathbf{t} \in \Omega$ and $q \in (0, 1)$ we have

$$\mathsf{AP}_i\left(\mathbf{t}, \omega, \mathsf{AP}_i^{-1}\left(\mathbf{t}, \omega, q\right)\right) = q \ . \tag{4}$$

Proposition 1 *Let* $\mathrm{NFA}_{max} \geqslant 0$.

(a) In the auto-similarity case, setting for all $\mathbf{t} \in \Omega$,

$$a(\mathbf{t}) = \mathsf{AP}_i^{-1}\left(\mathbf{t}, \omega, \mathrm{NFA}_{max}/|\Omega|\right) \ ,$$

we obtain that $\mathsf{ANFA}_i(\omega, a) = \mathrm{NFA}_{max}$.

(b) In addition, we have that for any $n \in \mathbb{N}\backslash\{0\}$

$$\mathbb{P}_0\left[\text{“at least } n \text{ offsets are detected in } U\text{”}\right] \leqslant \frac{\mathrm{NFA}_{max}}{n} \ .$$

In the template similarity case the same proposition holds if we replace $\mathsf{AP}_i(\mathbf{t}, \omega, a)$ by $\mathsf{TP}_i(v, \mathbf{t}, \omega, a)$.

Proof:

(a) The proof is done in the auto-similarity case and adapts to the template similarity framework. Using Equation (3), and $a(\mathbf{t}) = \mathsf{AP}_i^{-1}\left(\mathbf{t}, \omega, \mathrm{NFA}_{\max}/|\Omega|\right)$, we get

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

where the last equality is obtained using (4).

(b) Concerning the upper-bound, we have, using the Markov inequality, for any $n \in \mathbb{N}\backslash\{0\}$

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

□

Thus, setting a as in Proposition 1, in an image $u \in \mathbb{R}^{\Omega}$ an offset $\mathbf{t} \in \Omega$ is detected, *i.e.* yields spatial redundancy, for a given $\mathrm{NFA}_{\max}$ if

$$\mathcal{AS}_i(u, \mathbf{t}, \omega) \leqslant \mathsf{AP}_i^{-1}\left(\mathbf{t}, \omega, \mathrm{NFA}_{\max}/|\Omega|\right) \ . \tag{5}$$

This *a contrario* detection framework can then be simply rewritten as 1) computing the similarity function with input image u, 2) thresholding the obtained similarity map with the inverse cumulative distribution function of the computed similarity function under $\mathbb{P}_0$. The choice of $\mathbb{P}_0$ will be discussed in the next section. The computed threshold can depend on the offset and Proposition 1 ensures probabilistic guarantees on the expected number of detections under $\mathbb{P}_0$. In many cases, cumulative distribution function of similarity functions are easier to compute than inverse cumulative distribution function. Using the inverse property of the inverse cumulative distribution function and (5), we obtain that an offset is detected if and only if

$$\mathbb{P}_0\left[\mathcal{AS}_i(U, \mathbf{t}, \omega) \leqslant \mathcal{AS}_i(u, \mathbf{t}, \omega)\right] = \mathsf{AP}_i\left(\mathbf{t}, \omega, \mathcal{AS}_i(u, \mathbf{t}, \omega)\right) \leqslant \mathrm{NFA}_{\max}/|\Omega| \ . \tag{6}$$

This property will be used in Section 3.2 to define a similarity detection algorithm based on the evaluation of $\mathcal{AS}_{2,2}(u, \mathbf{t}, \omega)$ the auto-similarity associated to the square ℓ^2 norm, see Algorithm 1.

3 Gaussian model and detection algorithm

In this section we compute $\mathsf{AP}_i\left(\mathbf{t}, \omega, \alpha\right)$, *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. However this noise model does not contain any microtexture information. This can lead to the detection of many false positives, see Figure 10 for a tangible example where this problem occurs. Thus, depending on the application we may need a more general model that contains more structure. This model must fit two requirements:

- being computationally reasonable, so that the cumulative distribution function of its associated similarity functions can be numerically computed ;
- containing local spatial information.

A good proposal for this model is given by stationary Gaussian random fields defined in the following way: we introduce a spot, *i.e.* an image f over $\mathbb{R}^{\Omega}$ which contains the microtexture information we want to discard in our *a contrario* model. We suppose that f has zero spatial mean. We consider its associated microtexture model by defining, $U = f * W$, where $*$ is the convolution operator over defined by $u * v(\mathbf{x}) = \sum_{\mathbf{y}\in\Omega} \dot{u}(\mathbf{y})\dot{v}(\mathbf{x}-\mathbf{y})$ 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. The defined Gaussian random field has, in expectation, the same first and second-order moments as the original image f, *i.e.* for all $\mathbf{x}, \mathbf{y} \in \Omega$, we have

$$\mathbb{E}\left[U(\mathbf{x})\right] = 0 \qquad \text{and} \quad \operatorname{Cov}\left[U(\mathbf{x})U(\mathbf{y})\right] = \sum_{\mathbf{z}\in\Omega} \dot{f}(\mathbf{z})\dot{f}(\mathbf{z}+\mathbf{x}-\mathbf{y}) \; .$$

Thus in the following sections we are interested in the computation of the probability distribution function of similarity functions in Gaussian random fields.

3.1 Gaussian model

We consider a random field U over $\mathbb{Z}^2$ and compute local similarity measurements. An asymptotic approximation can be obtained when the patch size grows to infinity. In Theorem 1 we obtain a Gaussian asymptotic probability distribution in the auto-similarity case.

Theorem 1 *Let $(m_k)_{k\in\mathbb{N}}, (n_k)_{k\in\mathbb{N}}$ two positive increasing integer sequences and $(\omega_k)_{k\in\mathbb{N}}$ the sequence of subsets such that for any $k \in \mathbb{N}$, $\omega_k = [\![0, m_k]\!] \times [\![0, n_k]\!]$. Let $f \in \mathbb{R}^{\mathbb{Z}^2}$, $f \neq 0$ with finite support, W a Gaussian white noise over $\mathbb{Z}^2$ and $U = f * W$. For $i = sc$, p or (p,p), with $p \in (0, +\infty)$ there exist $\mu_i, \sigma_i \in \mathbb{R}^{\mathbb{Z}^2}$ and $(\alpha_{i,k})_{k\in\mathbb{N}}$ a positive sequence such that for any $\mathbf{t} \in \mathbb{Z}^2 \backslash \{\mathbf{0}\}$ we get*

(a) $\lim_{k\to+\infty} \frac{1}{\alpha_{i,k}} \mathcal{AS}_i(U, \mathbf{t}, \omega_k) \underset{a.s}{=} \mu_i(\mathbf{t})$;

(b) $\lim_{k\to+\infty} |\omega_k|^{\frac{1}{2}} \left(\frac{1}{\alpha_{i,k}} \mathcal{AS}_i(U, \mathbf{t}, \omega_k) - \mu_i(\mathbf{t}) \right) \underset{\mathcal{L}}{=} \mathcal{N}\left(0, \sigma_i(\mathbf{t})\right)$.

The proof can be found in the companion paper [42]. Similar asymptotic properties can be derived for template matching. In [42], constants are made explicit. Theorem 1 allows us to approximate the cumulative distribution function of some similarity function by the cumulative distribution function of a Gaussian random variable with given parameters. However, the generality of this theorem is not without compromise. Indeed, the approximation is valid only for large enough patch sizes. It is shown in [42] that due to the slow convergence towards the Gaussian distribution these approximations are not usable in practice, *i.e.* when the patch size is of the same order as the size of the spot support. In the following section we aim at deriving non-asymptotic approximation of the probability distribution function.

3.2 Detection algorithm

In this section Ω is a finite rectangular domain in $\mathbb{Z}^2$. We fix $\omega \subset \Omega$. We also define f, a zero-mean 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

$$\Delta_f(\mathbf{t}, \mathbf{x}) = 2\Gamma_f(\mathbf{x}) - \Gamma_f(\mathbf{x}+\mathbf{t}) - \Gamma_f(\mathbf{x}-\mathbf{t}) \; . \tag{7}$$

In the previous section we derived asymptotic properties for similarity functions. However, the asymptotic approximations are often not satisfying and non-asymptotic techniques are required. For instance, it should be noted that

the template matching for s_{sc} is Gaussian, thus explicit constants in Theorem 1 can be made exact for every patch size. The same remark holds for the template matching in the cosine case if the Gaussian model is a Gaussian white noise model. In what follows we restrict ourselves to the auto-similarity framework and consider the square of the ℓ^2 auto-similarity function, *i.e.* $\mathcal{AS}_{2,2}(u, \mathbf{t}, \omega)$. In this case we show that there exists an efficient method to compute the cumulative distribution function of the auto-similarity function.

The following proposition, proved in [42], 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 $\Omega = [\![0, M-1]\!]^2$ with $M \in \mathbb{N}$, $\omega \subset \Omega$, $f \in \mathbb{R}^\Omega$ and $U = f * W$ where W is a Gaussian white noise over Ω. The following equality holds for any $\mathbf{t} \in \Omega$*

$$\mathcal{AS}_{2,2}(U, \mathbf{t}, \omega) \underset{\mathcal{L}}{=} \sum_{k=0}^{|\omega|-1} \lambda_k(\mathbf{t}, \omega) Z_k \ , \tag{8}$$

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, see Equation (7)*, i.e for any $\mathbf{x_1}, \mathbf{x_2} \in \omega$, $C_{\mathbf{t}}(\mathbf{x_1}, \mathbf{x_2}) = \Delta_f(\mathbf{t}, \mathbf{x_1} - \mathbf{x_2})$.*

Note that in the special case where U is a standard Gaussian over $\mathbb{R}^\Omega$, *i.e.* a Gaussian white noise, and $(\mathbf{t} + \omega) \cap \omega = \varnothing$, we obtain that $C_{\mathbf{t}} = 2\,\mathrm{Id}$. Hence we get $\mathcal{AS}_{2,2}(U, \mathbf{t}, \omega) \underset{\mathcal{L}}{=} 2Z$, where Z 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 [43]. In [42] a projection method is introduced in order to easily compute approximated eigenvalues, with equality when $\omega = \Omega$. It performs a projection, for the Frobenius norm, of the covariance matrix $C_{\mathbf{t}}$ on the set of symmetric circulant matrices. The so-called Wood F method, see [44, 45], shows the best trade-off between accuracy and computational price 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. In what follows, we assume that we can compute the cumulative distribution function of $\mathcal{AS}_{2,2}(U, \mathbf{t}, \omega)$ and refers to [42] 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. We recall that, given a function a, an offset is detected if $\mathcal{AS}_i(u, \mathbf{t}, \omega) \leqslant a(\mathbf{t})$, see Definition 4. Using Proposition 1 and (6) that follows, we say that an offset is detected if $\mathsf{AP}_i\left(\mathbf{t}, \omega, \mathcal{AS}_i(u, \mathbf{t}, \omega)\right) \leqslant \mathrm{NFA}_{\max} / |\Omega|$. $\mathrm{NFA}_{\max}$ is supposed to be set by the user and we choose $i = (2, 2)$. In order to derive an auto-similarity detection algorithm we must precise our background model $\mathbb{P}_0$. This null hypothesis model will be built using only the second-order statistics of u. We first remove the spatial mean of u, *i.e.* we replace u by $u - \sum_{\mathbf{x} \in \Omega} u(\mathbf{x})/|\Omega|$, and consider the Gaussian random field given by $U = |\Omega|^{-1/2} u * W$, where W is a Gaussian white noise over Ω. Proposition 2 shows that choosing $\mathbb{P}_0$ to be the distribution of U we can compute $\mathsf{AP}_{2,2}(\mathbf{t}, \omega, \alpha)$ for any α, using the approximations mentioned and fully described in the previous paragraph, see [42] for more details.

Algorithm 1 Auto-similarity detection

function AUTOSIM-DETECTION(u, ω, $\mathrm{NFA}_{\max}$)
  $u \leftarrow u - \sum_{\mathbf{x} \in \Omega} u(\mathbf{x})/|\Omega|$
  for $\mathbf{t} \in \Omega$ **do**
    val $\leftarrow \mathcal{AS}_{2,2}(u, \mathbf{t}, \omega)$
    $P_{map}(\mathbf{t}) \leftarrow \mathsf{AP}_{2,2}(\mathbf{t}, \omega, \mathrm{val})$ ▷ $\mathsf{AP}_{2,2}(\mathbf{t}, \omega, \mathrm{val})$ approximation detailed above
    $D_{map}(\mathbf{t}) \leftarrow \mathbb{1}_{P_{map}(\mathbf{t}) \leqslant \mathrm{NFA}_{\max}/|\Omega|}$
  end for
  return the images P_{map}, D_{map}
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. [4] and was inspired by the work of Efros and Leung in texture synthesis [12]. It was also independently introduced in [46]. 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 which depend on this Euclidean distance. Usually the weight function is set to have exponential decay, but it was suggested in [47, 48, 49] to use compactly supported weight function in order to avoid the loss of isolated details. Since its introduction, many algorithm derived from NL-means have been proposed in order to embed the algorithm in general statistical frameworks [50, 41] or to take into account the underlying geometry of the patch space [51]. Among the state-of-the-art denoising algorithms, see [52] for a review, we consider Block-Matching and 3D Filtering (BM3D) [53] to compare our algorithm with.

There exist several works combining *a contrario* models and denoising tasks. Coupier et al. in [54] propose to combine morphological filters and a testing hypothesis framework to remove impulse noise. In [55] the authors 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 [56] and quasi-periodic noise [57], and rely on the thresholding of wavelet or Fourier coefficients.

We highlight the potential of the *a contrario* framework in redundancy detection by the statistical analysis of the NL-means denoising algorithm. Following a standard extension procedure of this algorithm we consider a threshold version of NL-means. In what follows we fix a "clean", or original, image u_0 defined over Ω, a finite rectangular domain over $\mathbb{Z}^2$, 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. We suppose the standard deviation to be known. Note that there exist several algorithms to estimate σ, see [58] 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

$$T = \left\{ \mathbf{t} \in \mathbb{Z}^2,\ \mathbf{t} + \omega \subset \Omega,\ \|\mathbf{t}\|_\infty \leqslant c \right\} \ , \tag{9}$$

with $c \in \mathbb{N}$. Note that the locality of the patch window was assessed to be a crucial feature of NL-means [59]. We now introduce our modification of NL-means.

Algorithm 2 NL-means threshold

function NL-MEANS-THRESHOLD(u, σ, ω_0, c, a)
  for $\mathbf{x} \in \mathbb{Z}^2,\ \mathbf{x} + \omega_0 \subset \Omega$ **do**
    $\omega \leftarrow \mathbf{x} + \omega_0$
    $T \leftarrow \left\{ \mathbf{t} \in \mathbb{Z}^2,\ \mathbf{t} + \omega \subset \Omega,\ \|\mathbf{t}\|_\infty \leqslant c \right\}$
    $N_\omega \leftarrow 0$
    $\hat{p}(u, \omega) \leftarrow 0$
    for $\mathbf{t} \in T$ **do**
      if $\mathcal{AS}_{2,2}(u, \mathbf{t}, \omega) \leqslant \sigma^2 a(\mathbf{t})$ **then** ▷ always true for $\mathbf{t} = 0$
        $\hat{p}(u, \omega) \leftarrow \frac{N_\omega}{N_\omega + 1} \hat{p}(u, \omega) + \frac{1}{N_\omega + 1} P_{\mathbf{t}+\omega}(u)$ ▷ recall that $P_{\mathbf{t}+\omega}(u) = u|_{\mathbf{t}+\omega}$
        $N_\omega \leftarrow N_\omega + 1$
      end if
    end for
  end for
  return $\hat{p}(u, \cdot)$
end function

Note Algorithm 2 returns denoised patches and not a denoised image. We obtain a pixel at position $\mathbf{x}$ in the denoised image $\hat{u}$ using the following average, see [60],

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

There exists a large literature concerning the setting of c and ω_0, see [49]. Note here that the output denoised version of the patch $\hat{p}(u,\omega)$ verifies the following equation

$$\hat{p}(u,\omega) = \sum_{\mathbf{t}\in T} \lambda_{\mathbf{t}} P_{\mathbf{t}+\omega}(u) \ , \qquad \lambda_{\mathbf{t}} = \frac{\mathbb{1}_{\mathcal{AS}_{2,2}(u,\mathbf{t},\omega)\leqslant a(\mathbf{t})}}{\sum_{\mathbf{s}\in T} \mathbb{1}_{\mathcal{AS}_{2,2}(u,\mathbf{s},\omega)\leqslant a(\mathbf{s})}} \ .$$

In the original NL-means method, $\lambda_{\mathbf{t}} = \frac{\exp\left(-\frac{\mathcal{AS}_{2,2}(u,\mathbf{t},\omega)}{h^2}\right)}{\sum_{\mathbf{t}\in T}\exp\left(-\frac{\mathcal{AS}_{2,2}(u,\mathbf{t},\omega)}{h^2}\right)}$. Setting h is not trivial and depends on many parameters (patch size, search window size, content of the original image). As in Algorithm 2 , we denote $N_\omega(u) = \sum_{\mathbf{t}\in T} \mathbb{1}_{\mathcal{AS}_{2,2}(u,t,\omega)\leqslant a(\mathbf{t})}$. The following proposition, similar to Proposition 1, gives a rationale for setting a.

Proposition 3 *Let $\varepsilon > 0$, $a \in \mathbb{R}^\Omega$ such that for any $\mathbf{t} \in \Omega$, $a(\mathbf{t}) = \mathsf{AP}_{2,2}^{-1}(\mathbf{t},\omega,1-\varepsilon)$, with background model a Gaussian white noise W, see Section 2.2. Let T be defined in* (9) *and $N_\omega(W) \in \{0,\dots,T\}$ the random number of selected patches used to denoise patch $P_\omega(W)$, see Algorithm 2. Then for any $n \in \mathbb{N}\backslash\{0\}$ it holds that*

$$\mathbb{P}_0\left[|T| - N_\omega(W) \geqslant n\right] \leqslant \frac{|T|\varepsilon}{n} \ .$$

Proof: Using the Markov inequality, we have

$$\mathbb{P}_0\left[|T| - N_\omega(W) \geqslant n\right] \leqslant \frac{|T| - \sum_{\mathbf{t}\in T}\mathbb{E}\left[\mathbb{1}_{\mathcal{AS}_{2,2}(W,\mathbf{t},\omega)\leqslant a(\mathbf{t})}\right]}{n} \leqslant \frac{|T|\varepsilon}{n} \ .$$

□ While Proposition 3 is very similar to Proposition 1, there exists a crucial difference in the underlying *a contrario* model. In Proposition 1, the rare event considered in the noise model is the high similarity between two patches. Here, the rare event considered is the high dissimilarity between two patches. Note that this dissimilarity detection in Gaussian white noise images is close to the anomaly detection model proposed by Davy et al. in [26]. In their model, an offset is said to be detected, *i.e.* to be a false alarm, if there is no spatial redundancy in the background model. To be more precise an offset $\mathbf{t} \in T$ is said to be detected if

$$\mathcal{AS}_2(u,\mathbf{t},\omega) > a(\mathbf{t}) \ .$$

The quantity $|T| - N_\omega(W)$ represents the number of patches that are *not* used to denoise patch $P_\omega(W)$, *i.e.* the number of detected offsets. In order to denoise a standard Gaussian white noise we want this number to be small, *i.e.* $N_\omega(W)$ to be close to $|T|$. Considering this *a contrario* framework, Proposition 3 rewrites

$$\mathbb{P}_0\left[\text{“at least } n \text{ offsets are detected in } W\text{”}\right] \leqslant \frac{|T|\varepsilon}{n} \ .$$

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. Thus by setting $\varepsilon = \mathrm{NFA}_{\max}/|T|$ we obtain, as in Proposition 1, that $\mathsf{ANFA}_{2,2}(\omega,a)$ is equal to $\mathrm{NFA}_{\max}$. 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 Gaussian white noise over Ω, $u_0 \in \mathbb{R}^\Omega$ and $\sigma > 0$. Let $\omega = \mathbf{x} + \omega_0$ be a fixed patch and let $\varepsilon > 0$. We introduce the random set $\hat{T} = \{\mathbf{t} \in T,\ \mathcal{AS}_{2,2}(U,\mathbf{t},\omega) \leqslant \sigma^2 a(\mathbf{t})\}$ (the selected offsets) with $a(\mathbf{t}) = \mathsf{AP}_{2,2}^{-1}(\mathbf{t},\omega,1-\varepsilon)$ as in Proposition 3 and T defined in* (9)*. Let $a_T = \max_{\mathbf{t}\in T} a(\mathbf{t})$. Then for any $a_W > 0$, setting $\varepsilon_W = 1 - \mathbb{P}\left[\|P_\omega(W)\|_2^2 \leqslant a_W \mid \hat{T}\right]$, we have*

$$\mathbb{P}\left[\|\hat{p}(U,\omega) - P_\omega(u_0)\|_2 \leqslant \sigma(a_T^{1/2} + a_W^{1/2}) \mid \hat{T}\right] \geqslant 1 - \varepsilon_W \ . \tag{10}$$

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

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

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

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

We also have that by definition of ε_W

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

▫ Note that this upper-bound depends on two quantities: a_T which is related to the number of similar neighbors and a_W which corresponds to the loss of information due to the addition of noise. Choosing $\varepsilon = 1$, which implies $a_T = 0$, leads to only one patch identified to be similar with $P_\omega(u)$: the patch itself. Doing so, we strongly overfit the noise.

In order to implement Algorithm 2 we need to compute $a(\mathbf{t}) = \mathsf{AP}_{2,2}^{-1}(\mathbf{t},\omega,1-\varepsilon)$ with a Gaussian white noise background model. We recall that in Section 3.2 we give a method to compute this quantity in general Gaussian settings. In Proposition 2, we highlighted that the probability distribution function of $\mathcal{AS}_{2,2}(W,\mathbf{t},\omega)$ is given as a weighted sum of chi-square random variables with parameter 1. We shall use the Wood F approximation mentioned in Section 3.2 in order to compute its inverse cumulative distribution function. Since the spot$f$ is equal to δ_0, the Dirac function in 0, in the case of a Gaussian white noise, our case of interest here, we can compute directly the eigenvalues and do not need any approximation, see the following proposition.

Proposition 5 *Let $\mathbf{t} = (t_x, t_y) \in \mathbb{Z}^2\backslash\{0\}$, $C_\mathbf{t}$ as in Proposition 2 with spot $f = \delta_0$ and $\omega = [\![0, p-1]\!]^2$, with $p \in \mathbb{N}$. If $\|\mathbf{t}\|_\infty \geqslant p$ then $C_\mathbf{t} = 2\,\mathrm{Id}$ else, supposing that $t_x \neq 0$, we have, expressing $C_\mathbf{t}$ in the basis corresponding to the raster scan order on the x-axis*

$$C_\mathbf{t} = \begin{pmatrix} B_0 & B_1 & \dots & B_{p-1} \\ B_1^T & B_0 & \ddots & \vdots \\ \vdots & \ddots & B_0 & B_1 \\ B_{p-1}^T & \dots & B_1^T & B_0 \end{pmatrix}.$$

with for any $\ell \in [\![0, p-1]\!]$, $B_\ell \in \mathcal{M}_p(\mathbb{R})$ and $B_\ell = 0$ except if $\ell = |t_x|$ in which case $B_{|t_x|} = -D_{|t_y|}$ where D_j is a zero matrix with ones on the j-th diagonal[1] with $j \in [\![-(p-1), p-1]\!]$, or if $\ell = 0$ in which case $B_0 = 2\,\mathrm{Id}$. 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 $m \in [\![2, q+1]\!]$, $k \in [\![1, m-1]\!]$ and $q = \lceil \frac{p}{|t_x| \vee |t_y|} \rceil$. For any $m \in [\![2, q+1]\!]$, $k \in [\![1, m-1]\!]$ it holds

(a) for any $k' \in [\![1, m-1]\!]$, $r_{m,k} = r_{m,k'}$;

(b) $r_{m,k} = 2|t_x||t_y|$ if $2 \leqslant m < q$;

(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 way. 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 2 we illustrate that $a(\mathbf{t})$ found using the proposed method is nearly constant for fixed patch size (8×8) and patch search window (21×21). Thus in our implementation we suppose that $a(\mathbf{t})$ 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. In Figure 3 we present a first comparison with the NL-means algorithm. Perceptual results as well as Peak Signal to Noise Ratio (PSNR) measurements [2] are commented. Results on other images than Barbara are displayed in Figure 4.

[1] If $j \geqslant 0$ the we consider the $|j|$-th upper diagonal. If $j \leqslant 0$ we consider the $|j|$-th lower diagonal.

[2] $\mathrm{PNSR}(u,v) = 10\log_{10}\left(\frac{\max_\Omega u^2}{\|u-v\|_2^2}\right)$.

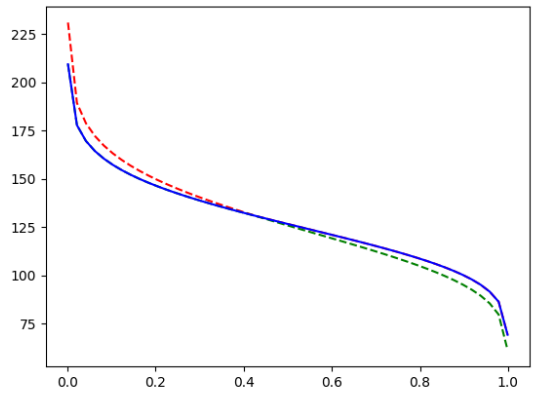


Figure 2: ***Thresholds dependence in*** ε**.** In this figure we display $a(\mathbf{t}) = \mathsf{AP}_{2,2}^{-1}(\mathbf{t}, \omega, 1-\varepsilon)$ as a function of ε. The patch size is fixed to 8×8, the offsets $\mathbf{t}$ satisfy $\|\mathbf{t}\|_\infty \leqslant 10$. The red dashed line is given 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 that patches are independent, *i.e.* $a(\mathbf{t}) = 2\chi_{64}^{-1}(1-\varepsilon)$. The maximal increase between the maximum of $a(\mathbf{t})$ and the minimum of $a(\mathbf{t})$ is of 13.0%.

If the threshold is high, *i.e.* $\varepsilon \ll 1$ then almost no patch is rejected. In consequence, the output denoised image is very smooth. This smoothness is a correct guess for constant patches. However, this proposition does not hold when the region contains details. Indeed, in this case details are lost due to the averaging process. By setting a conservative threshold, *i.e.* $\varepsilon \approx 0.1$, for example, we reject all the patches for which the structure does not strongly match the one of the input patch, see Figure 5. 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 [53], solve this problem by treating this case as an exception, applying a specific denoising method in this situation. For instance, the patch can be denoised by hard-thresholding its spectral coefficients. Note that in the original BM3D algorithm [53] the ℓ^2 distance is not computed between the images directly but between coarse-filtered versions of the patches. If our model can easily be extended to any linear transformation of the patches, the prefiltering applied in BM3D is not linear and thus a preprocessed Gaussian patch does not have a Gaussian probability distribution function anymore. We show the differences between our version of NL-means and BM3D in Figure 6 .

In Figure 7, we show that Algorithm 2 has a good behavior compared to the original NL-means algorithm. By setting $\varepsilon = 0.01$ 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

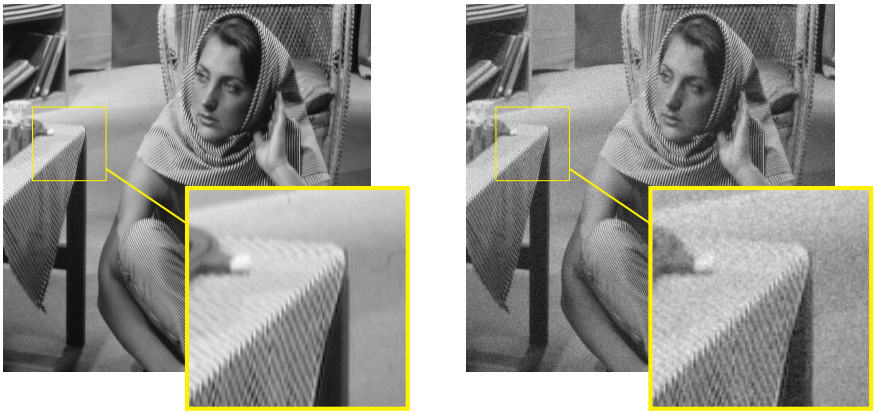
5.1 Existing algorithms

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.

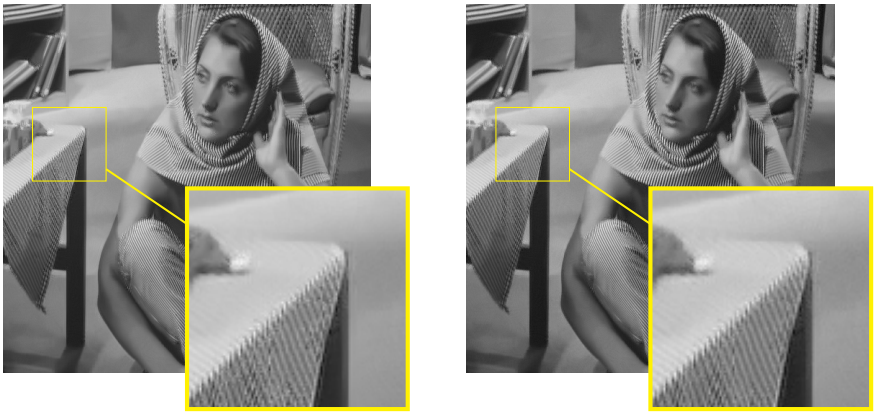
Periodicity detection is a long-standing problem in texture analysis [62]. First algorithms used the quantization of images, relying on co-occurrence matrices and statistical tools like χ_2 tests or κ tests. Global methods extract peaks in the frequency domain (Fourier spectrum) [35] or in the spatial domain (autocorrelation).

In [34] the notion of inertia is introduced. It is defined for any $\mathbf{t} \in \Omega$ by

$$\mathcal{I}(\mathbf{t}) = \sum_{(i,j)\in[\![0,N_g]\!]^2} (i-j)^2 \left(\sum_{\mathbf{z}\in\Omega} \mathbb{1}_{u(\mathbf{z})=i} \mathbb{1}_{u(\mathbf{z}+\mathbf{t})=j} \right),$$



(a) (b)



(c) PSNR = 29.81 (d) PSNR = 29.29

Figure 3: ***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 $\varepsilon = 0.01$, 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). PSNR and perceptual results are almost the same for both algorithms.

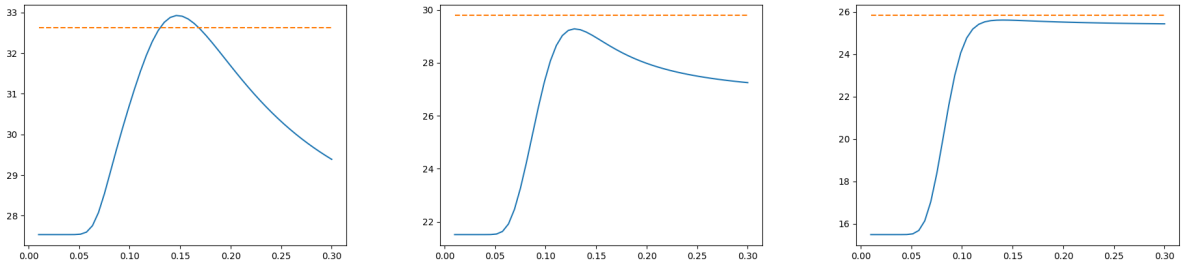
Figure 4: ***NL-means comparison.*** In this figure we compare Algorithm 2 with the traditional NL-means algorithm. Once again ω_0 is fixed to be a 8×8 square. The first column contains clean images, the second column represents the same images corrupted by a Gaussian noise with $\sigma = 20$. The third column is the output of Algorithm 2 with ε fixed to 0.01 and the last column is the output of the NL-means algorithm for the optimal value of h (with regards to the PSNR). Perceptual results and PSNR are comparable, even though our algorithm yields slightly better PSNR values.

(a) $\varepsilon = 0.2$ (b) $\varepsilon = 0.1$ (c) $\varepsilon = 0.01$

Figure 5: ***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 one, *i.e.* the patch is not denoised. As ε decreases the number of detected offsets increases. Note that $|\hat{T}|$ is small for textured regions with contrast changes and regions with details.

(a) (b) (c) (d)

Figure 6: ***Comparison with BM3D.*** We compare Algorithm 2 to BM3D [53]. 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 ouput of BM3D, with default parameters, see [61]. The result in (c) corresponds to the output of Algorithm 2 with $\varepsilon = 0.01$. The output (c) is too blurry compared to (b). In order to correct this behavior we set $\varepsilon = 0.1$, *i.e.* increase the global threshold. Some improvements are noticeable, for examples the two marks on the left leg are retrieved in (b) and (d) but not in (c). However the image remains blurry and artifacts due to the overfitting of the noise appear, this is known as the *rare patch effect* in [50]. For instance, some patches in the scarf are not denoised anymore.



(a) $\sigma = 10$ (b) $\sigma = 20$ (c) $\sigma = 40$

Figure 7: PSNR ***study.*** In this figure we present the evolution of the Peak Signal to Noise Ratio (PSNR) for different values of h 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 $\varepsilon = 0.01$. 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.

where u is a quantized image on $N_g + 1$ gray levels. The following proposition generalizes this definition and extends to a local framework results from [63].

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 introduce the ω-inertia for any $\mathbf{t} \in \Omega$*

$$\mathcal{I}_{\omega}(\mathbf{t}) = \sum_{(i,j)\in[\![0,N_g]\!]^2} (i-j)^2 \left(\sum_{\mathbf{z}\in\omega} \mathbb{1}_{\dot{u}(\mathbf{z})=i} \mathbb{1}_{\dot{u}(\mathbf{z}+\mathbf{t})=j} \right) .$$

Then we have

$$\mathcal{I}_{\omega}(\mathbf{t}) = \mathcal{AS}_{2,2}(u, \mathbf{t}, \omega) .$$

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

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

□

If $\omega = \Omega$ then the ω-inertia statistics is exactly the inertia introduced in [34] and the result is due to [63]. Proposition 6 gives an insight on the specificity of the $\mathcal{AS}_{2,2}(u, \mathbf{t}, \omega)$ statistics: it relies on patches and local features instead of global image properties, thus allowing for more flexibility in the periodicity analysis, at the cost of selecting the patch size and position in the original image.

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 [39] uses a recursive framework which consists in 1) a lattice model proposal based on detectors such as Kanade-Lucas-Tomasi (KLT) feature trackers [64], 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 main difference with [39] is that our algorithm is rigid to deformations since it does not follow a recursive scheme and does not include a spatial warping step. 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 sparse graph. We use the same notation for the detection mapping $\mathbf{t} \mapsto \mathbb{1}_{\mathcal{AS}_i(u,\mathbf{t},\omega)\leqslant a(\mathbf{t})}$ *i.e.* the D_{map} output of Algorithm 1, which is a binary function over the offsets, and the set of detected offsets. The graph $\mathscr{G} = (V, E)$ is then built as follows:

- **Vertices**: for each 8-connected component $\mathscr{C}_k$ in D_{map} we note $\mathbf{v}_k$ one position for which the minimum of $\mathcal{AS}_{2,2}(u, \mathbf{t}, \omega)$ over $\mathscr{C}_k$ is achieved. The set of vertices V is defined as $V = (\mathbf{v}_k)_{k\in[\![1,N_{\mathscr{C}}]\!]}$ where $N_{\mathscr{C}}$ 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. This assumption was proposed and its limitations studied in [39]. This defines a set of edges E.

Refering to the three steps of [39] 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 $\mathscr{G} = (V, E)$ be a random graph with $V \subset \mathbb{R}^2$. We say that $\mathscr{G}$ 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*

$$\mathbf{e} = m_\mathbf{e} b_1 + n_\mathbf{e} b_2 + \sigma Z_\mathbf{e},$$

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 $(m_\mathbf{e}, n_\mathbf{e})_{\mathbf{e}\in E} \in \mathbb{Z}^{2|E|}$.

This definition can be translated into a model on the vertices of the graph but the independence hypothesis would not hold anymore.

The parameters in this model are given by (B, M, σ^2). The parameters (B, M) describe the structure of $\mathscr{G}$ as a subset of a lattice with basis B whereas σ describes the validity of the model. Indeed, if $\max(\|b_1\|_2, \|b_2\|_2) \approx \sigma$ then there is too much variability in the set of edges for $\mathscr{G}$ to be perceived as a lattice. Since a lattice is characterized by its basis vectors, performing inference in this graphical model will allow us to retrieve the underlying lattice, if present in the original detection set. In the following we compute the Maximum Likelihood Estimator (MLE) in this model, *i.e.* we maximize the likelihood in the parameters (B, M, σ^2) given the observations.

In the context of pixels, edges are integer-valued vectors since images are defined over finite discrete grid and a trivial solution for the MLE is then given by $B = ((0, 1), (1, 0))$, see Figure 8. In order to avoid this overfitting situation we propose *a priori* probability distribution function on parameters B and M

- $\pi_1(B) \propto \exp\left(-\frac{\delta_B}{2\sigma^2}\|B\|_2^2\right)$, with $\delta_B > 0$.
- $\pi_2(M) \propto \exp\left(-\frac{\delta_M}{2\sigma^2}\|M\|_2^2\right)$, with $\delta_M > 0$.

Note that instead of the Gausian distributions on B and M we could have chosen only a hard constraint on B, given for instance by $\pi_0(B) \propto \mathbb{1}_{\|b_1\|\geqslant\beta}\mathbb{1}_{\|b_2\|\geqslant\beta}\mathbb{1}_{\sin(b_1,b_2)\geqslant\gamma}$, with $\beta, \gamma > 0$. This constraint may be more natural to introduce but is encoded as an improper prior which is difficult to include in a MLE computation. On the other hand, the Gaussian assumption corresponds to a soft constraint on the coefficients. Note that the behavior enforced by π_0 and π_2 is the same since imposing that coefficients are not too large is equivalent to imposing that basis vectors are not too small. The resulting *a priori* distribution based on the two Gaussian constraints is denoted by π

$$\pi(B, M) \propto \exp\left(-\frac{1}{2\sigma^2} r(B, M)\right), \text{ with } r(B, M) = \delta_B\|B\|_2^2 + \delta_M\|M\|_2^2 . \tag{11}$$

We use the distribution presented in (11) as an *a priori* in our model. The log-likelihood of the full model on the edges E is given by

$$\mathscr{L}(E|B, M, \sigma^2) = -2(|E| + 1)\log(\sigma^2) - \frac{1}{2\sigma^2}\underbrace{\left(\sum_{\mathbf{e}\in E}\|m_\mathbf{e} b_1 + n_\mathbf{e} b_2 - \mathbf{e}\|^2 + r(B, M)\right)}_{q(E|B,M)} . \tag{12}$$

A discussion on the dependence of the model on the hyperparameters (δ_B, δ_M) is conducted in Figure 8. Finding the MLE of this full log-likelihood 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. Indeed let $q(\mathbf{v}) = (\mathbf{v}-\mathbf{v_0})^T S(\mathbf{v}-\mathbf{v_0})$, with S a positive definite symmetric matrix, $\mathbf{v}_0 \in \mathbb{Z}^2$ and $\mathbf{v} \in \mathbb{Z}^2$. We have

$$\min_{\mathbf{v}\in\mathbb{Z}^2} (\mathbf{v}-\mathbf{v_0})^T S(\mathbf{v}-\mathbf{v_0}) = \min_{\mathbf{y}\in S^{1/2}(\mathbb{Z}^2-\mathbf{v_0})} \|\mathbf{y}\|_2^2 \,.$$

This last formulation is known as the Shortest Vector Problem (SVP) which is a challenging problem [65] (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. Suppose that S is diagonal and $\mathbf{v}_0 = (x_0, y_0)$, then the optimal solution over $\mathbb{Z}^2$ is given by $([x_0], [y_0])$ where $[\cdot]$ is the nearest integer operator. Hence the rounding approximation is exact if the matrix S is diagonal. We will see in Proposition 7 a condition for this situation to occur.

Algorithm 3 Lattice detection – Alternate minimization

function ALTERNATE-MINIMIZATION(E, δ_B, δ_M, N_{it})
  $M_0 \leftarrow 0$
  $B_0 \leftarrow$ initialization procedure ▷ initialization discussed in Section 5.2
  for $n \leftarrow 0$ to $N_{it}-1$ **do**
    $\tilde{M} \leftarrow \underset{M\in\mathbb{R}^{2|E|}}{\operatorname{argmin}}\, q(E|B_n, M)$ ▷ expression given in Proposition 7
    if $q\left(E|B_n, [\tilde{M}]\right) < q(E|B_n, M_n)$ **then**
      $M_{n+1} \leftarrow [\tilde{M}]$
    else
      $M_{n+1} \leftarrow M_n$
    end if
    $B_{n+1} \leftarrow \underset{B\in\mathbb{R}^4}{\operatorname{argmin}}\, q(E|B, M_{n+1})$ ▷ expression given in Proposition 7
  end for
  $\sigma^2_{N_{it}} \leftarrow \underset{\sigma^2\in\mathbb{R}_+}{\operatorname{argmin}}\, -\mathscr{L}(E|B_{N_{it}}, M_{N_{it}}, \sigma^2)$
  return $B_{N_{it}}, M_{N_{it}}, \sigma^2_{N_{it}}$
end function

For any $\sigma > 0$ we denote by $\mathscr{L}_n(\sigma) = \mathscr{L}(E|B_n, M_n, \sigma^2)$, with $n \in \mathbb{N}$, the log-likelihood sequence. This algorithm is closely related to other matrix factorization algorithms [66].

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

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

with $\otimes$ the tensor product between matrices and

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

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

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

Proof: $(\mathscr{L}_n(\sigma))_{n\in\mathbb{N}}$ is non-decreasing by construction because

$$\mathscr{L}_n(\sigma) \leqslant \mathscr{L}(E|B_n, M_{n+1}, \sigma^2) \leqslant \mathscr{L}_{n+1}(\sigma)\,.$$

Let us show that the sequences $(M_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ are bounded. Since $(\mathscr{L}_n(\sigma))_{n\in\mathbb{N}}$ is non-decreasing, the sequence $\big(q(E|B_n, M_n, \sigma^2)\big)_{n\in\mathbb{N}}$ is non-increasing. We obtain that

$$\delta_M \|M_n\|^2 \leqslant q(E|B_0, M_0, \sigma^2) \ , \qquad \delta_B \|B_n\|^2 \leqslant q(E|B_0, M_0, \sigma^2) \ .$$

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 $n \geqslant n_0 + 1$, there exists $n_1 \in \mathbb{N}$, with $n_1 \geqslant n$ such that $M_{n_1} = M_{n_0}$ thus

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

Hence for every $n \geqslant n_0 + 1$, $\mathscr{L}_n(\sigma) = \mathscr{L}(E|B_n, M_n, \sigma^2) = \tilde{\mathscr{L}}(\sigma)$. Suppose there exists $n \geqslant n_0 + 1$ such that $M_n \neq M_{n+1}$ this means that $\mathscr{L}(E|B_n, M_{n+1}, \sigma^2) > \mathscr{L}_n(\sigma)$ (because of lines 6 and 7 of Algorithm 3) which is absurd. Thus $(M_n)_{n\in\mathbb{N}}$ is stationary and so is $(B_n)_{n\in\mathbb{N}}$. ▫

It must be noted that this alternate minimization algorithm can be written as an Expectation Maximization (EM) algorithm [67] where Gaussian distributions are approximated as a Dirac in their modes. Indeed, knowing M, σ^2 and E, the basis B follows a Gaussian distribution. Hence we can perform an estimation of M using the EM framework. This procedure introduces a hierarchy among the parameters. Proposition 8 gives the new update rule. Note that replacing Σ_{EM} by 0 we obtain the update rule of Proposition 7.

Proposition 8 (EM **update rule)** *Conditionally to (E, M, σ), B has a Gaussian probability distribution function denoted π_{EM} with mean $\mu_{EM} \in \mathbb{R}^4$ and covariance $\Sigma_{EM} \in M_4(\mathbb{R})$ such that*

$$\Sigma_{EM} = \sigma^2 \left(\Lambda_M \otimes \mathrm{Id}_2\right)^{-1} \ , \qquad \mu_{EM} = \left(\Lambda_M \otimes \mathrm{Id}_2\right)^{-1} E_M \ .$$

If line 5 in Algorithm 3 is replaced by the EM estimation $\tilde{M} = \underset{M\in\mathbb{R}^{2|E|}}{\operatorname{argmin}} \mathbb{E}_{EM}\left[q(E, B, M)\right]$, then we have

$$\tilde{M} = \left(\mathbb{E}_{\pi_{EM}}\left[\Lambda_B\right] \otimes \mathrm{Id}_{|E|}\right)^{-1} E_{B_n} = \left(\left(\Lambda_{B_n} + 2\sigma_n^2 \Lambda_{M_n}^{-1}\right) \otimes \mathrm{Id}_{|E|}\right)^{-1} E_{B_n} \in \mathbb{R}^{2|E|} \ .$$

Proof: The proof is postponed to Appendix B. ▫

Notice that since Σ_{EM} depends on σ, the estimation of σ can no longer be conducted at the end of the algorithm but instead becomes sequential.

In Algorithm 3 as well as in its EM extension, 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. Other initialization procedures are implemented such as orthonormal (up to a dilatation factor) direct basis where the first vector is given by the longest edge or random initialization on the circle with radius given by the mean norm of edges.

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 9. 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 [44]. Lattice detection is performed using Algorithm 3 with parameters $\delta_M = 10$ and $\delta_B = 10^{-2}$.

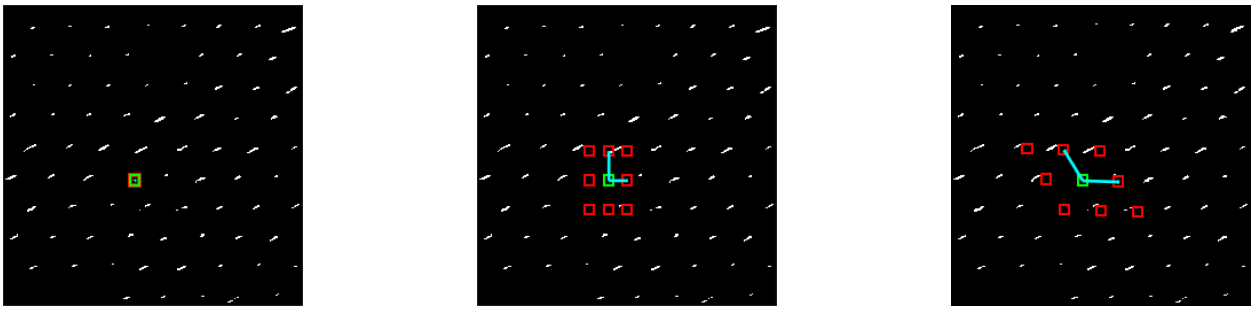
5.3.1 Escher paving

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.



(a) $\delta_M = 0$ $\delta_B = 0$ (b) $\delta_M = 5$ $\delta_B = 10^{-1}$ (c) $\delta_M = 9$ $\delta_B = 10^{-1}$

Figure 8: ***Influence of coefficient 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.

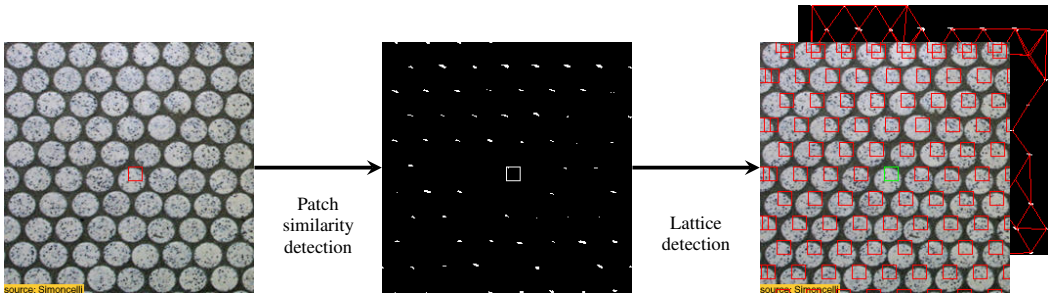


Figure 9: ***Lattice proposal algorithm.*** Lattice detection and extraction in images require a patch from the user and compute a binary image containing all the offsets with correct similarity as well as a lattice matching the underlying graph. The patch auto-similarity detection step was presented 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. Indeed, our similarity detection method follows an *a contrario* framework and thus, the identified patches reveal similarity which is not likely to be found in the background model. Hence the more structured is the background noise model the less we obtain detections. This situation is considered in Figure 10. In this experiment we underline that choosing a Gaussian white noise random field as a background model does not yield enough structure in order to obtain robustness in the setting of $\mathrm{NFA}_{\mathrm{max}}$.

$\mathrm{NFA}_{\mathrm{max}}$ 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, $\mathrm{NFA}_{\mathrm{max}}$ must be set carefully otherwise two situations can occur:

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

(b) if $\mathrm{NFA}_{\mathrm{max}}$ is too low, we fail to identify important perceptual structures in the image.

We observed that a general good practice is to set $\mathrm{NFA}_{\mathrm{max}}$ equal to 10, see Figure 11. However, if the input patch is corrupted one may increase this parameter up to 10^2 or 10^3, see Figure 16 and Figure 17.

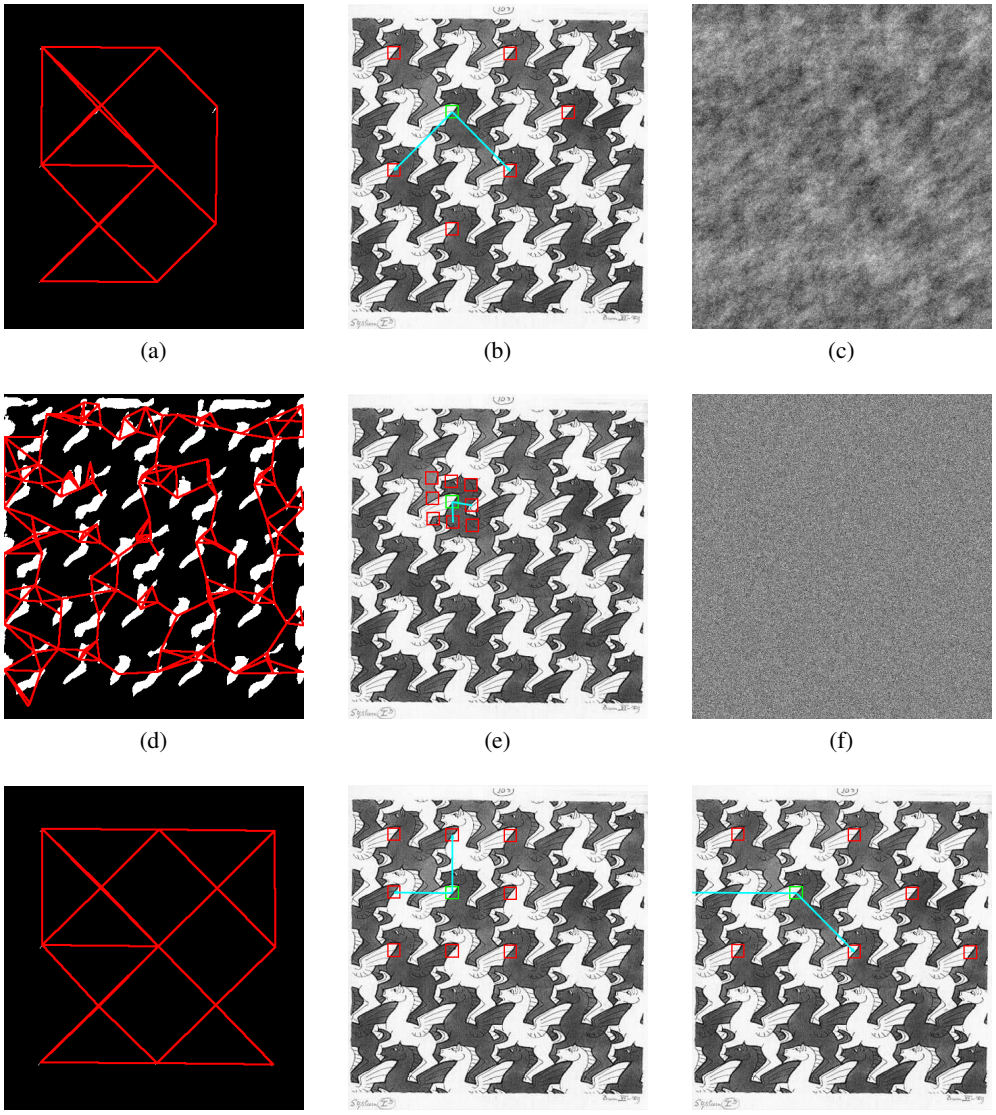
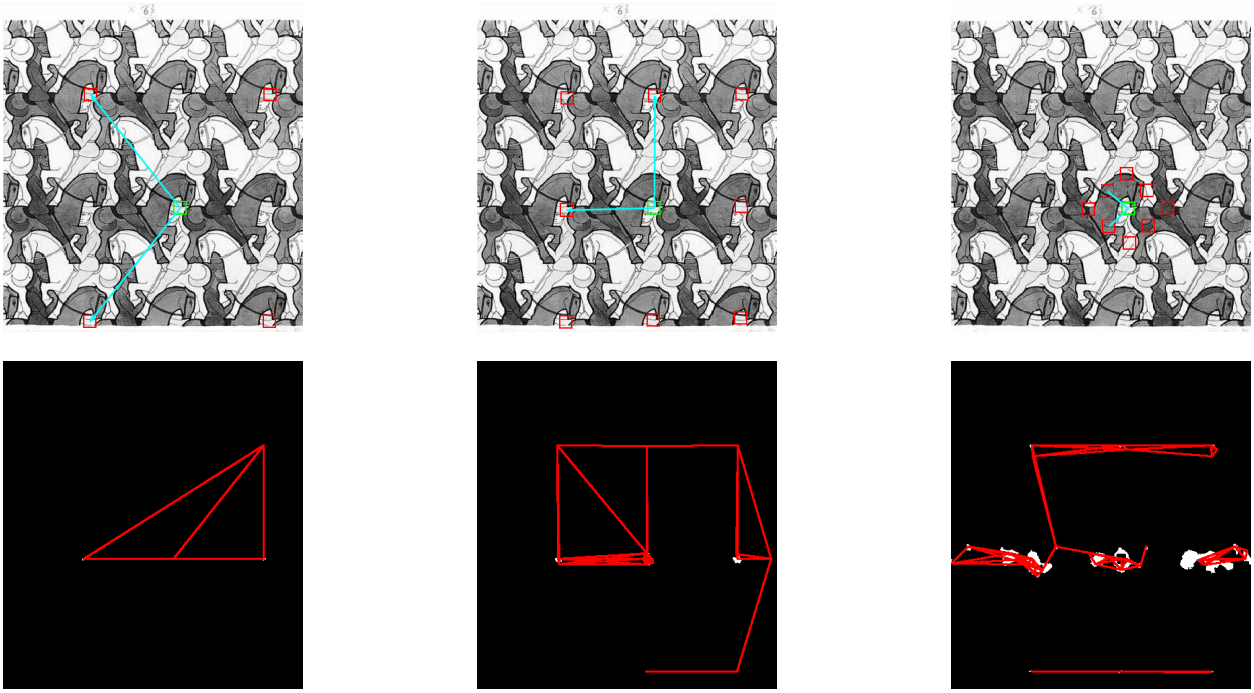


Figure 10: ***Choice of the microtexture model.*** In this experiment we discuss the choice of the *a contrario* background microtexture model. In the left column we display the graph obtained after the detection step. In the middle column we superpose the proposed lattice on the original image. The 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 a Gaussian model with a spot equal to the full image and $\mathrm{NFA}_{\mathrm{max}}$ is set to 10. A sample of this Gaussian 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) $\mathrm{NFA}_{\mathrm{max}}$ is set to 10, thus allowing in average 10 false alarms, *i.e.* detected offsets, in Gaussian white noise. 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 instead of more structured microtexture model, we must set the $\mathrm{NFA}_{\mathrm{max}}$ 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. Indeed the perceptual lattice is a sub-lattice of the perceptual lattice given by the arrangement of the white horses. This situation is corrected in (i) with random initialization for B_0. In (h) final log-likelihood value is -565.5 which is inferior to the final log-likelihood value in (i): -542.1. Thus (i) gives a better local maximum of the full log-likelihood than (h).



(a) (b) (c)

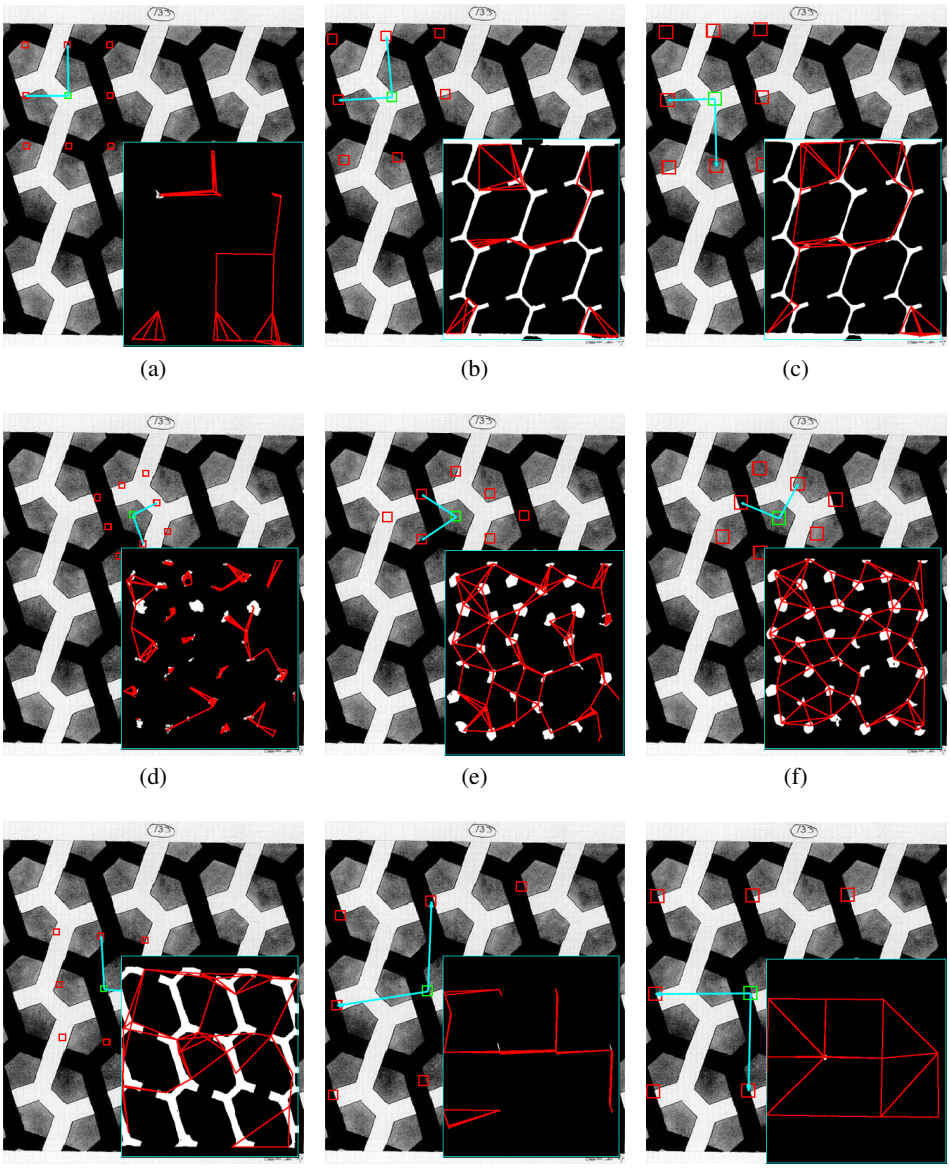
Figure 11: ***Choice of Number of False Alarms.*** In this experiment we discuss the choice of the $\mathrm{NFA}_{\max}$ parameter in the *a contrario* framework in the case where the underlying microtexture model is a Gaussian random field with full image as a spot. Each column corresponds to a pair of images: the returned lattice and its associated underlying graph. In (a), $\mathrm{NFA}_{\max}$ is set to 1. Detections are correct, there are not enough points to precisely retrieve the perceptual lattice and the algorithm returns a lattice which is contained in the perceptual one. This problem could be avoided starting with another initialization in Algorithm 3. In (b,) $\mathrm{NFA}_{\max}$ is set to 10. The estimated lattice is correct. In (c), $\mathrm{NFA}_{\max}$ 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.

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 12 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 [37].

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 locally 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 can be automated [37]. In Figure 13, 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 global methods such as analyzing the peaks of the modulus of the Fourier transform of the image would have failed here since the structure is local. 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, $\mathrm{NFA}_{\max}$. This situation is illustrated in Figure 14.

Finally we assess the precision of our measurements by comparing our results with a model used in crystallography, see Figure 15. 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 (12).

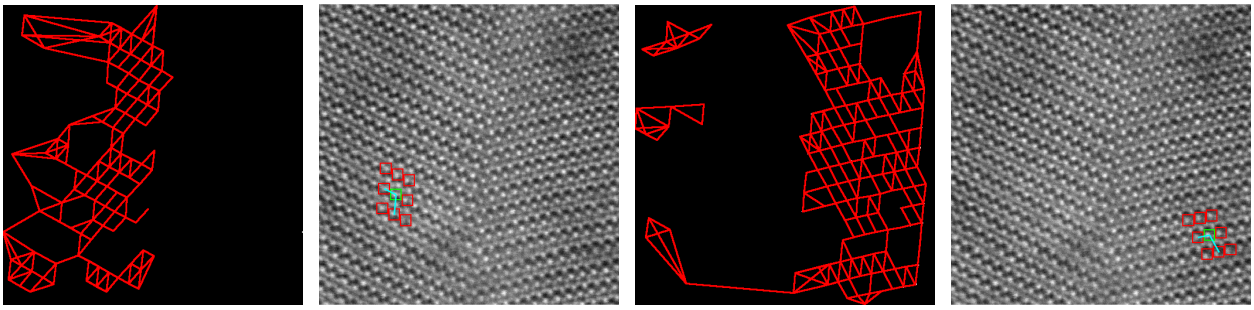


(a) (b) (c)

(d) (e) (f)

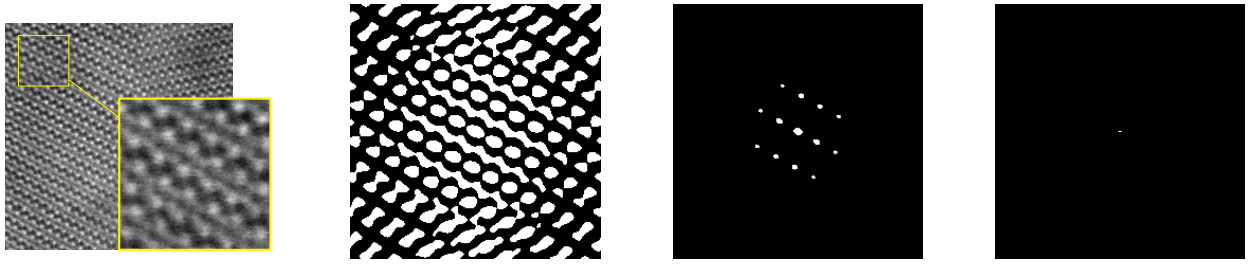
(g) 10×10 (h) 15×15 (i) 20×20

Figure 12: ***Influence of patch size and patch position.*** For each experiment $\mathrm{NFA}_{\mathrm{max}}$ is set to 10^4, *i.e.* 4 % of the pixels. In most cases lower $\mathrm{NFA}_{\mathrm{max}}$ could be used but setting a high $\mathrm{NFA}_{\mathrm{max}}$ 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 where all the white regions are detected as similar ((b) and (c)). On the second row the patch contains gray microtexture which has some local structure. Once again this structure is less considered as the patch size grows. 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 even if there are some errors since the white area is small. In (i), the patch size is large enough to cover large areas of the three gray levels and the perceptual lattice is identified.



(a) (b) (c) (d)

Figure 13: ***Lattice extraction.*** In this experiment we consider a crystallographic image (an orthorombic NiZr alloy) and set $\mathrm{NFA}_{\max}$ 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.



(a) (b) 90% (c) 95% (d) 99%

Figure 14: ***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).

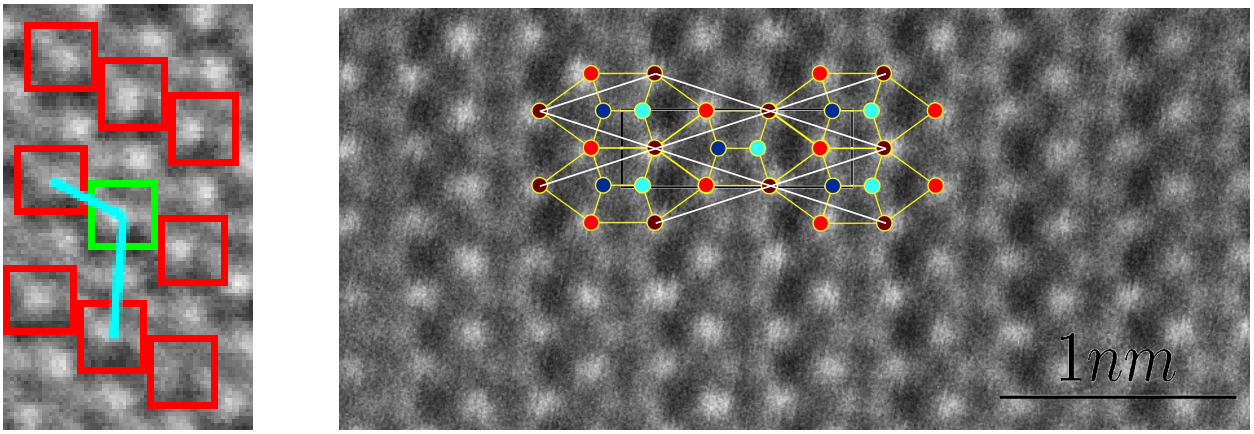
5.3.3 Natural images

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 16, and the effect of the camera position, see Figure 17.

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 spot f then $h * U$ is a sample from a Gaussian model with spot $h * f$. Thus all the properties derived earlier remain valid with this linear operation. In Figure 16, we set h to be a Laplacian operator [3]. 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

[3] We use a 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.



(a) (b)

Figure 15: ***Agreement with crystallography models.*** In (a) we perform a zoom on of the lattice identified in Figure 13 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 parallelogramm. Up to rotation these basis vectors match the one identified in (a). However, the parallelogramm 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.

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 17.

6 Texture ranking

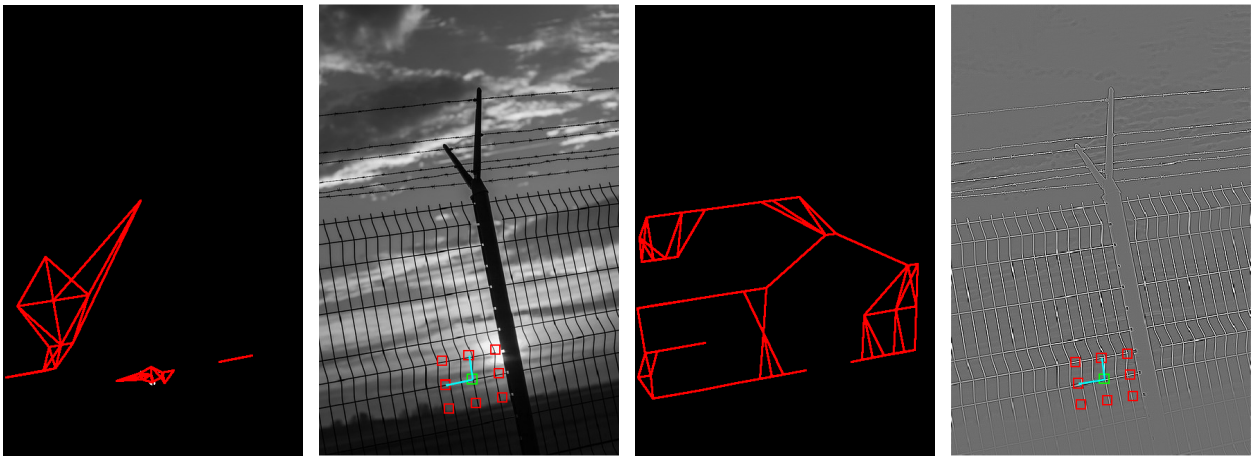
We conclude these experiments by showing that this simple graphical model can be used to perform ranking among texture images, sorting them according to their degree of periodicity. We say that an image has high periodicity degree if a lattice structure can be well fitted to the image. We introduce a criterion for evaluating the relevance of the lattice hypothesis, and thus assessing the degree of periodicity of the texture. Let u be an image over Ω, let $\omega \subset \Omega$ be a patch domain and a be as in Proposition 1 with $\mathrm{NFA}_{\max}$ set by the user.

Definition 6 (Periodicity criterion) *Let $\{\mathbf{t} \in \Omega,\ \mathcal{AS}_{2,2}(u, \mathbf{t}, \omega) \leqslant a(\mathbf{t})\}$ be the set of detected offsets and $N_{\mathscr{C}}$ its number of connected components as defined in Section 5.2. Let also $(\widehat{B}, \widehat{M}, \widehat{\sigma})$ be the estimated parameters using Algorithm 3. We define the following periodicity criterion c_{per} as*

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

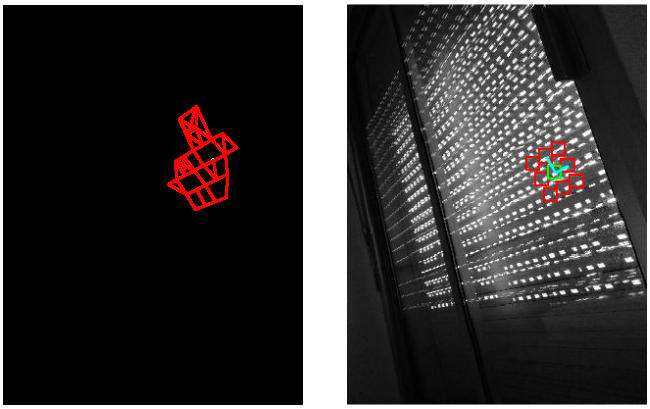
where $\widehat{B} = (\hat{b}_1, \hat{b}_2)$.

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



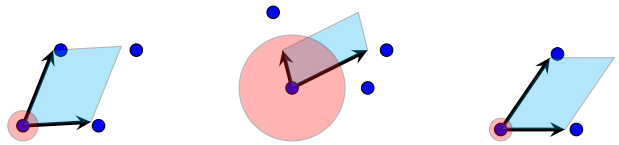
(a) (b) (c) (d)

Figure 16: ***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), $\mathrm{NFA}_{\max}$ was set to 10^5 which corresponds to 35 % of detection in the associated *a contrario* model. Lower $\mathrm{NFA}_{\max}$ 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 $\mathrm{NFA}_{\max}$ 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.



(a) (b)

Figure 17: ***Homography and locality.*** In this experiment $\mathrm{NFA}_{\max}$ was set to 10^3. Note that the detected graph is localized around the original patch. This leads to a satisfying lattice proposal in a small neighborhood around the original patch. However this proposal is not valid for the whole image.



(a) (b) (c)

Figure 18: ***Parameters configurations.*** In each figure, dark blue points correspond to the vertex of the graph $\mathscr{G}$. Arrows represent the estimated basis $\widehat{B}$ and its associated parallelogram is displayed in blue. Red disks have radius $\widehat{\sigma}$. In (a) there exists a lattice in the graph which corresponds to some perceptual lattice on the vertices. In this case $\frac{\pi\widehat{\sigma}^2}{|\det(\hat{b}_1,\hat{b}_2)|}$, the ratio of the uncertainty area and the primitive area, is small. In (b) no perceptual lattice is identified onto the vertices and the estimated lattice vectors are not valid. This is expressed with a high ratio $\frac{\pi\widehat{\sigma}^2}{|\det(\hat{b}_1,\hat{b}_2)|}$. In (c), the graph $\mathscr{G}$ contains only three vertices. Thus the lattice approximation is nearly optimal (up to regularization factors) and the ratio $\frac{\pi\widehat{\sigma}^2}{|\det(\hat{b}_1,\hat{b}_2)|}$ is very small. However, three points are not perceptually identified as a lattice. Hence considering c_{per} we take into account the number of points and we may retrieve that (a) is considered more periodic than (b) and (c).

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 dependence 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, see Figure 18.

This c_{per} criterion is arbitrary and many others could be derived. For example we could consider for any $\lambda > 0$

$$c_{per}^{\lambda} = \frac{\pi\widehat{\sigma}^2}{N_{\mathscr{C}}^{\lambda}|\det(\hat{b}_1,\hat{b}_2)|},$$

Taking the logarithm of this criterion we obtain for any $\lambda > 0$

$$\log\left(c_{per}^{\lambda}\right) = \log\left(\frac{\pi\widehat{\sigma}^2}{|\det(\hat{b}_1,\hat{b}_2)|}\right) + \lambda\log\left(\frac{1}{N_{\mathscr{C}}}\right),$$

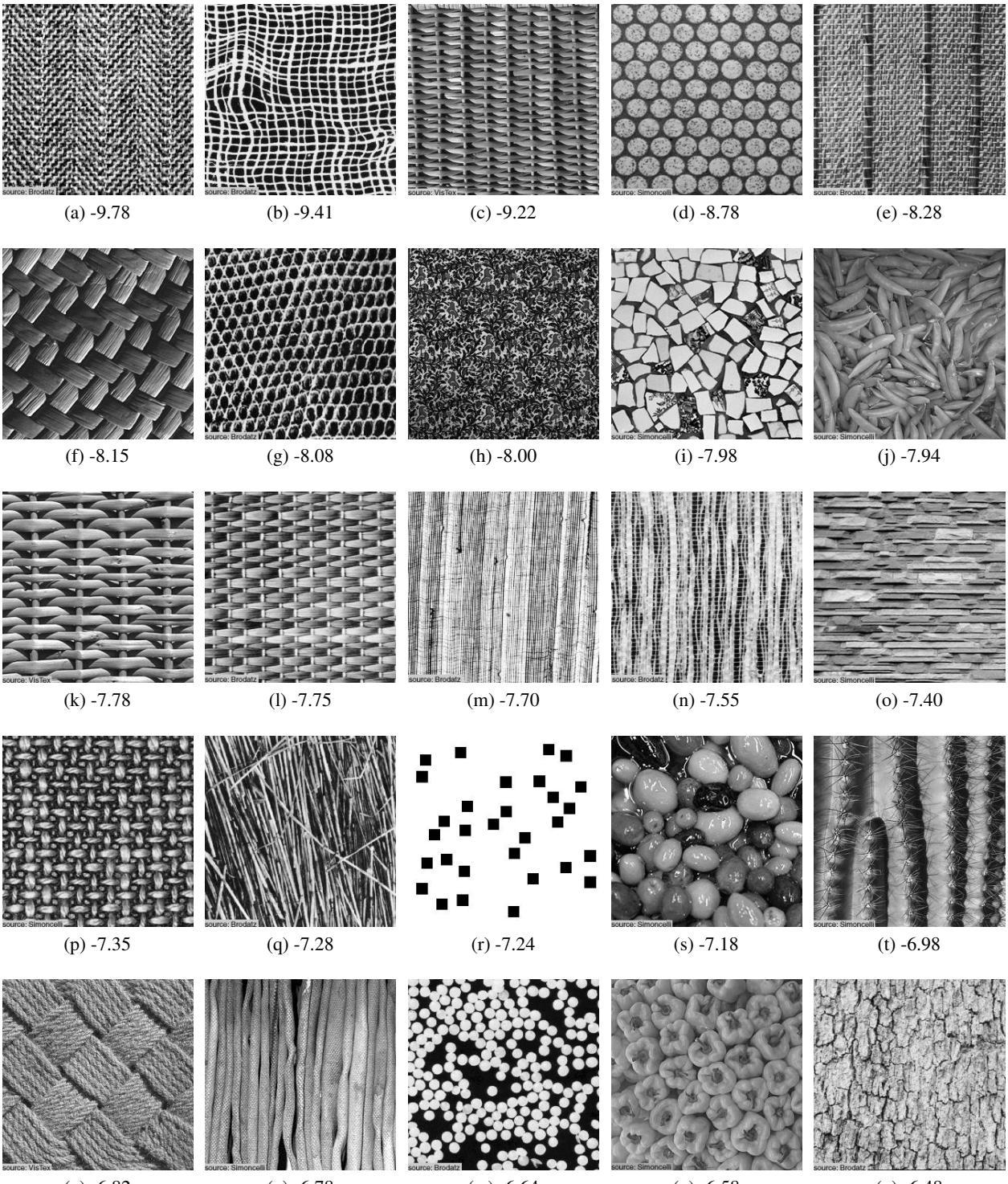
and λ can be interpreted as a trade-off parameter between the area penalty and the number of detection penalty. We present the result of the texture ranking algorithm on 25 images of the Brodatz dataset, see Figure 19. The criterion classifies most of the texture correctly except for (i), (j) and (p). Notice that it is easy, from a perceptual point of view, to decide if a texture is periodic or not: from (a) to (l) all images can be labeled as periodic and from (m) to (y) all images can be labeled as non-periodic. However intra-class ranking is difficult.

7 Conclusion

In this paper we introduce a statistical model, the *a contrario* framework, to analyze spatial redundancy in images. We present similarity functions in the context of image processing. After conducting a study on the statistical properties of these similarities measures 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. Doing so, we aim at giving a better understanding of some image processing algorithms in which rules-of-thumb prevail in parameter selection.

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 [68] or to take into account the geometry of the patch space [51, 69].

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



(a) -9.78 (b) -9.41 (c) -9.22 (d) -8.78 (e) -8.28

(f) -8.15 (g) -8.08 (h) -8.00 (i) -7.98 (j) -7.94

(k) -7.78 (l) -7.75 (m) -7.70 (n) -7.55 (o) -7.40

(p) -7.35 (q) -7.28 (r) -7.24 (s) -7.18 (t) -6.98

(u) -6.82 (v) -6.78 (w) -6.64 (x) -6.58 (y) -6.48

Figure 19: ***Texture ranking.*** In this experiment 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 30 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 $\text{NFA}_{\text{max}} = 1$, $\delta_M = 10$, $\delta_B = 10^{-2}$ and $N_{it} = 10$. The c_{per} criterion is computed for each setting. We associate to each image the median of the 30 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.

defined on a graph. Proving the finite-time convergence of Algorithm 3 we study its links with classical Expectation-Maximization formulation. 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 $\mathrm{NFA}_{\max}$ settings. The lattice extraction algorithm is used on art images such as Escher pavings. 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 [40]. Finally our method is tested on natural images where some of its limits such as perspective defect or sensitivity to occlusion phenoma 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* 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 [17]. Another potential direction is to look at the behavior of the introduced similarity functions for more general random fields in order to handle more complex and structured situations such as parametric texture synthesis.

8 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

We start with the proof of Proposition 5.

Proof: 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 $J : \Omega_0 \to [\![2, +\infty[\![$ such that for any $\mathbf{x}_0 \in \Omega_0$

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

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

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

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 $(\Lambda_{\mathbf{x}_0,k})_{k \in [\![1, J(\mathbf{x}_0)-1]\!]}$ is in the kernel of the matrix

$(\sin(\ell k\pi/(J(\mathbf{x}_0)-1)))_{1\leqslant j,\ell\leqslant J(\mathbf{x}_0)-1}$. Since the sinus discrete transform is invertible we obtain that for any $\mathbf{x}_0\in\Omega_0$ and $k\in[\![1,J(\mathbf{x}_0)-1]\!]$, $\Lambda_{\mathbf{x}_0,k}=0$. 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

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

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 $J(\Omega_0)\subset[\![2,q+1]\!]$. Let $m\in[\![2,q-1]\!]$. The cardinality of $K_{k,m}$ is the cardinality of $J^{-1}(m)$. Let $\mathbf{x}_0\in\Omega_x$ we have

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

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

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

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

We have computed $|K_{k,m}|$ for every $m\in[\![2,q-1]\!]$. 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 $\Omega_x\cap J^{-1}(q+1)=\Omega_y\cap J^{-1}(q+1)=\varnothing$. Let $\mathbf{x}_0\in\Omega_{x,y}$ then $\mathbf{x}_0=-\mathbf{t}+(x,y)$ with $x\in[\![0,t_x-1]\!]$ and $y\in[\![0,t_y-1]\!]$. We obtain the following equivalence

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

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

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

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. □

B Update rules

We derive the proof of Proposition 7.

Proof: Computing the minimum of $q(E|B,M)$ 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

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

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

We give the proof of Proposition 8.

Proof: Since

$$q(E|B,M)=\|\left(\Lambda_M\otimes\mathrm{Id}_2\right)^{\frac{1}{2}}\left(B-\left(\Lambda_M\otimes\mathrm{Id}_2\right)^{-1}E_M\right)\|^2+\alpha(M)\ ,$$

we obtain that

$$\mathscr{L}(E|B,M,\sigma^2)=-\frac{1}{2\sigma^2}\|\left(\Lambda_M\otimes\mathrm{Id}_2\right)^{\frac{1}{2}}\left(B-\left(\Lambda_M\otimes\mathrm{Id}_2\right)^{-1}E_M\right)\|^2+\beta(\sigma,M)\ ,$$

which proves that conditionally to E, M and σ^2, B is Gaussian with mean $\left(\Lambda_M\otimes\mathrm{Id}_2\right)^{-1}E_M$ and covariance matrix $\sigma^2\left(\Lambda_M\otimes\mathrm{Id}_2\right)^{-1}$. □

References

[1] 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.

[2] 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.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–24:11, 2009.

[4] 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 2005), 20-26 June 2005, San Diego, CA, USA*, pages 60–65, 2005.

[5] 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 1991, Providence, RI, USA, April 27-30, 1991*, pages 309–318, 1991.

[6] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random phase textures: Theory and synthesis. *IEEE Trans. Image Processing*, 20(1):257–267, 2011.

[7] 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. 2015USPCB041.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999.

[13] 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 2001, Los Angeles, California, USA, August 12-17, 2001*, pages 341–346, 2001.

[14] 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.

[15] 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.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[17] 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.

[18] 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.

[19] 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.

[20] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, Nov 2003.

[21] 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.

[22] George R. Cross and Anil K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5(1):25–39, 1983.

[23] 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.

[24] 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, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1902–1910, 2016.

[25] J. Bruna and S. Mallat. Multiscale Sparse Microcanonical Models. *ArXiv e-prints*, January 2018.

[26] 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.

[27] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

[28] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Edge Detection by Helmholtz Principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.

[29] 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.

[30] 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.

[31] Agnes Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt theory to image analysis: a probabilistic approach*, volume 34. Springer Science & Business Media, 2007.

[32] David G. Lowe and Thomas O. Binford. Perceptual organization as a basis for visual recognition. In *Proceedings of the National Conference on Artificial Intelligence, Washington, D.C., USA, August 22-26, 1983.*, pages 255–260, 1983.

[33] Song Chun Zhu. Embedding Gestalt laws in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1170–1187, 1999.

[34] 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.

[35] 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.

[36] 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.

[37] Niklas Mevenkamp and Benjamin Berkels. Unsupervised and accurate extraction of primitive unit cells from crystal images. In *Pattern Recognition - 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*, pages 105–116, 2015.

[38] Xiahan Sang and James M. LeBeau. Revolving scanning transmission electron microscopy: Correcting sample drift distortion without prior knowledge. *Ultramicroscopy*, 138:28 – 35, 2014.

[39] 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.

[40] 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.

[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] Valentin De Bortoli, Agnes Desolneux, Bruno Galerne, and Arthur Leclaire. Redundancy in Gaussian random fields. *Submitted*, Nov 2018.

[43] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48:419–426, 1961.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] Joseph Salmon. On two parameters for denoising with non-local means. *IEEE Signal Process. Lett.*, 17(3):269–272, 2010.

[49] Vincent Duval, Jean-François Aujol, and Yann Gousseau. On the parameter choice for the Non-Local Means. 37 pages., March 2010.

[50] Vincent Duval, Jean-François Aujol, and Yann Gousseau. A bias-variance approach for the nonlocal means. *SIAM J. Imaging Sciences*, 4(2):760–788, 2011.

[51] Antoine Houdard, Charles Bouveyron, and Julie Delon. High-Dimensional Mixture Models For Unsupervised Image Denoising (HDMI). working paper or preprint, June 2017.

[52] M. Lebrun, M. Colom, A. Buades, and J. M. Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] Frédéric Sur. An a-contrario approach to quasi-periodic noise removal. In *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, pages 3841–3845, 2015.

[58] David L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Information Theory*, 41(3):613–627, 1995.

[59] 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.

[60] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-Local Means Denoising. *Image Processing On Line*, 1:208–212, 2011.

[61] Marc Lebrun. An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line*, 2:175–213, 2012.

[62] 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.

[63] Gyuhwan Oh, Seungyong Lee, and Sung Yong Shin. Fast determination of textural periodicity using distance matching function. *Pattern Recognition Letters*, 20(2):191–197, 1999.

[64] 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 '81, Vancouver, BC, Canada, August 24-28, 1981*, pages 674–679, 1981.

[65] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2001.

[66] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the Netflix prize problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008*, pages 267–274, 2008.

[67] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977. With discussion.

[68] 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.

[69] Yi-Qing Wang and Jean-Michel Morel. SURE guided gaussian mixture image denoising. *SIAM J. Imaging Sciences*, 6(2):999–1034, 2013.