

A Non-Local Dual-Domain Approach to Cartoon and Texture Decomposition

Frédéric Sur

Université de Lorraine, LORIA UMR CNRS 7503, INRIA projet Magrit

Abstract

This paper addresses the problem of cartoon and texture decomposition. Microtextures being characterized by their power spectrum, we propose to extract cartoon and texture components from the information provided by the power spectrum of image patches. The contribution of texture to the spectrum of a patch is detected as statistically significant spectral components with respect to a null hypothesis modeling the power spectrum of a non-textured patch. The null-hypothesis model is built upon a coarse cartoon representation obtained by a basic yet fast filtering algorithm of the literature. Hence the term “dual domain”: the coarse decomposition is obtained in the spatial domain and is an input of the proposed spectral approach. The statistical model is also built upon the power spectrum of patches with similar textures across the image. The proposed approach therefore falls within the family of non-local methods. Experimental results are shown in various application areas, including canvas pattern removal in fine arts painting, or periodic noise removal in remote sensing imaging.

Keywords: *Cartoon and texture decomposition, dual-domain algorithm, non-local estimation, Fourier domain.*

This is the author-manuscript version of

F. Sur. A Non-Local Dual-Domain Approach to Cartoon and Texture Decomposition. *IEEE Transactions on Image Processing*, 2019.

DOI: 10.1109/TIP.2018.2881906

The final authenticated version is available online at:
<https://doi.org/10.1109/TIP.2018.2881906>

1 Introduction

Decomposing an image as the sum of geometric and textural components is a problem of image analysis that has lead to a great deal of work. In this problem, known as cartoon and texture decomposition, the cartoon component is piecewise smooth, made of the geometric shapes of the images, and the texture component is made of stationary or quasi-stationary oscillatory patterns filling the shapes. Of course, the problem, based on perceptual clues, is intrinsically ill-posed and any definition of cartoon and texture is subject to discussion. After an overview of the literature, we introduce a new algorithm for cartoon and texture decomposition based on a dual-domain approach [1]: a spatial domain filtering gives a coarse decomposition, which permits estimating statistics of the power spectrum of cartoon patches. Estimating these statistics is based on the similarity of textured patches across the image, in the same spirit as the non-local approaches [2]. Therefore, the resulting algorithm performs a non-local dual-domain cartoon and texture decomposition.

1.1 Related works

Most authors define the cartoon component as a bounded-variation function whose total variation is controlled, which discards oscillatory patterns and enforces its piecewise constant nature, giving cartoon components with sharp edges. In contrast to the cartoon component, the texture component is the subject of discussions and a huge literature has emerged in the last decades. Variational approaches aim at decomposing the input image as the sum of the cartoon and texture component by minimizing some energy, often defined as the sum of the total variation (or some variant of it) of the cartoon part and of some norm of the texture part. This latter norm may be defined as the L^2 -norm [3], L^1 -norm [4], Meyer's G -norm [5, 6], Gabor-norm [7], or norms enforcing a sparse representation in an adapted basis [8], just to cite a few. The authors of [9] address the problem of structured noise, which can be seen as a texture covering the whole image domain. Structured noise is indeed modeled as a stationary process and a variational approach to denoising is proposed in [9]. Recently, it has been proposed [10] to model the texture part as the sum of a small number of base textures in a low patch-rank approach. According to the authors of [11], low patch-rank outperforms previous approaches which smooth the cartoon component by transferring high-frequency information from edges to the texture instead of the cartoon component. It is, however, a global approach which is not suited to natural images where textures show different patterns of different nature. A local approach based on the block nuclear norm is proposed in [11]. The interested reader is asked to refer to the complete discussion in [7, 10, 11, 12].

All these methods are slow, because it is difficult to find a numerical solution of the above-mentioned variational problems. A fast non-linear filter for cartoon and texture decomposition is proposed by the authors of [12] (software available in [13]). It is based on the relative reduction rate of the local total-variation across scale, which is a good feature to measure whether an image point belongs to a textured region or not. This spatial filter giving quite a blurry cartoon part, a follow-up paper [14] (software in [15]) makes use of banks of filters adapted to image edges. A scale factor governs the trade-off between textures and shapes, large-size texture details being considered as shapes rather than texture. Recent papers investigate the use of a guidance signal for image filtering [16, 17, 18] (after [19]), which makes it possible to remove small-scale details without affecting other structures. Applications of this approach include texture removal, the difference with the aforementioned methods is probably that the retrieved cartoon component is quite sketchy.

Many of these works are based on the fact that textures are well represented in the Fourier

domain. It is indeed well known that microtextures are characterized by their power spectrum, as discussed in, e.g., [20] concerning texture feature extraction, and in [21], after [22], for texture rendering. The power spectrum of a texture even generally reduces to a few spikes of various amplitudes, which justifies the sparse representation of [8, 10]. Some authors make use of such a property to design notch filter to remove structured noise covering the whole image domain [23, 24, 25, 26, 27]. In these papers, the difficulty is to automatize the detection of texture spikes. It is proposed in [26] to detect texture spikes as deviation of the expected power spectrum of natural images, which decreases proportionally to some power of the inverse of the frequency [28]. The expected power spectrum is simply obtained by averaging patches covering the whole image as in [24], which is not possible when considering localized textures. The method was adapted to a-contrario detection in [27]. Removing canvas patterns in fine art paintings can also be considered as a form of textured noise removal. For instance, the authors of [29] remove canvas patterns in paintings by fitting a parametric model of the canvas to the texture part given by the algorithm of [12].

1.2 Contribution

The proposed contribution consists in locally detecting the texture components as statistically significant Fourier coefficients under a null hypothesis modeling non-textured patches. Our statistical model is based on two ingredients: first, the Gaussian nature of image patches, which is the ground of several recent papers on denoising [30, 31, 32, 33]; second, the use of a coarse cartoon and texture decomposition which permits estimating statistics of non-textured patches in a non-local way.

Section 2 gives the expectation and variance of the power spectrum components of a random Gaussian non-textured patch. Since the probability distribution of patches is not constant across the whole image domain, these statistics are empirically estimated in a non-local way. We select a sample of patches with similar power spectra and extract the cartoon parts with the coarse yet fast algorithm of [12, 13] in order to estimate the above-mentioned statistics. This approach is in line with the famous non-local means methodology [2] in which similar patches are assumed to be independent realizations of the same random patch.

Section 3 explains how to build a statistical test to detect texture spectrum spikes in each image patch, based on the null model of Section 2. A non-linear filter separating cartoon and texture components is thus designed, and implemented using the discrete windowed Fourier transform.

Since texture is extracted in the Fourier domain from a coarse information coming from a spatial filter, this is a dual-domain approach, a term coined by the authors of [1] when proposing to refine denoising in the frequency domain after a first estimation in the spatial domain. Note that the present contribution is not, strictly speaking, a dual-domain filter in the sense of [1].

Experiments are discussed in Section 4. In order to reproduce the experiments presented here, we make a Matlab code available on the following webpage, together with a companion document showing additional experiments:

<https://members.loria.fr/FSur/software/NoLoDuDoCT/>

1.3 Notation and reminder

The Fourier transform of any image patch p defined over the pixel domain $\mathcal{D}_p = \{1, \dots, L\} \times \{1, \dots, L\}$ is denoted by \hat{p} and is such that for any $\boldsymbol{\xi} = (\xi, \eta) \in \{1, \dots, L\} \times \{1, \dots, L\}$,

$$\hat{p}(\boldsymbol{\xi}) = \sum_{\mathbf{x} \in \mathcal{D}_p} p(\mathbf{x}) e^{-2i\pi \langle \mathbf{x}, \boldsymbol{\xi} \rangle} \quad (1)$$

where $\mathbf{x} = (x, y) \in \mathcal{D}_p$ and $\langle \mathbf{x}, \boldsymbol{\xi} \rangle = (x\xi + y\eta)/L$.

The modulus and conjugate of any complex number z are denoted by $|z|$ and \bar{z} . The power spectrum of a patch p is $|\hat{p}|^2$.

The expectation and variance of a random variable V are denoted by $E(V)$ and $\text{var}(V)$, respectively. The covariance of two random variables V and V' is denoted by $\text{cov}(V, V')$. The empirical estimation of a statistic S is denoted by \tilde{S} .

We recall that if n is a 0-mean noise patch of variance $\sigma^2(\mathbf{x})$ at pixel \mathbf{x} , then the Fourier transform of \hat{n} satisfies, for any (ξ, η) , $E(\hat{n}(\xi, \eta)) = 0$, and $E(|\hat{n}(\xi, \eta)|^2) = \sigma^2$ where $\sigma^2 = \sum_{\mathbf{x} \in \mathcal{D}_p} \sigma^2(\mathbf{x})$ is the total variance calculated over the patch domain.

2 The power spectrum of Gaussian patches

The Fourier transform of a Gaussian patch follows a complex normal distribution, and the power spectrum coefficients follow a non-central generalized chi-square distribution. While accurate approximations of this distribution are available [34, 35], their computational cost is not suitable for our use. It is a standard trick to use the Gaussian approximation which is more tractable and justified by the central limit theorem.

2.1 Expectation and variance of the power spectrum

We assume that any random patch $p_0 = c + n$ is the sum of a random non-textured component c and a Gaussian noise n of variance $\sigma^2(\mathbf{x})$ at pixel \mathbf{x} , statistically independent from $c(\mathbf{x})$. Since we aim at building a statistical model for the absence of texture (null hypothesis), we assume that p_0 does not show any texture, although this does not impact the calculation of the expectation and variance in this section.

The Fourier transform being linear, $\hat{p}_0 = \hat{c} + \hat{n}$, and the power spectrum of the patch p thus writes, for any frequency $\boldsymbol{\xi} = (\xi, \eta)$:

$$|\hat{p}_0(\boldsymbol{\xi})|^2 = |\hat{c}(\boldsymbol{\xi})|^2 + |\hat{n}(\boldsymbol{\xi})|^2 + \hat{c}(\boldsymbol{\xi})\overline{\hat{n}(\boldsymbol{\xi})} + \overline{\hat{c}(\boldsymbol{\xi})}\hat{n}(\boldsymbol{\xi}) \quad (2)$$

We now calculate the expectation and variance of $|\hat{p}_0(\boldsymbol{\xi})|^2$, for any $\boldsymbol{\xi}$. For better readability, we drop $\boldsymbol{\xi}$ in the following equations.

2.1.1 Expectation

Since \hat{c} and \hat{n} are independent, $E(\hat{n}) = 0$, and $E(|\hat{n}|^2) = \sigma^2$, the following equation holds:

$$\begin{aligned} E(|\hat{p}_0|^2) &= E(|\hat{c}|^2) + E(|\hat{n}|^2) + E(\hat{c}\overline{\hat{n}}) + \overline{E(\hat{c})}E(\hat{n}) \\ &= E(|\hat{c}|^2) + \sigma^2 \end{aligned} \quad (3)$$

2.1.2 Variance

Variance algebra gives:

$$\begin{aligned} \text{var}(|\hat{p}_0|^2) &= \text{var}(|\hat{c}|^2) + \text{var}(|\hat{n}|^2) + \text{var}(\hat{c}\overline{\hat{n}} + \overline{\hat{c}}\hat{n}) \\ &\quad + 2\text{cov}(|\hat{c}|^2, |\hat{n}|^2) + 2\text{cov}(|\hat{c}|^2, \hat{c}\overline{\hat{n}} + \overline{\hat{c}}\hat{n}) + 2\text{cov}(|\hat{n}|^2, \hat{c}\overline{\hat{n}} + \overline{\hat{c}}\hat{n}) \end{aligned} \quad (4)$$

We successively calculate:

- $\text{cov}(|\hat{c}|^2, |\hat{n}|^2) = 0$ since \hat{c} and \hat{n} are independent.
- $\text{cov}(|\hat{c}|^2, \widehat{c\hat{n}} + \widetilde{c\hat{n}}) = E((|\hat{c}|^2 - E(|\hat{c}|^2))(\widehat{c\hat{n}} + \widetilde{c\hat{n}})) = E((|\hat{c}|^2 - E(|\hat{c}|^2))\widehat{c})\overline{E(\hat{n})} + E((|\hat{c}|^2 - E(|\hat{c}|^2))\widetilde{c})E(\hat{n}) = 0$ since \hat{c} and \hat{n} are independent and $E(\hat{n}) = 0$.
- $\text{cov}(|\hat{n}|^2, \widehat{c\hat{n}} + \widetilde{c\hat{n}}) = E((|\hat{n}|^2 - E(|\hat{n}|^2))(\widehat{c\hat{n}} + \widetilde{c\hat{n}})) = E(|\hat{n}|^2\widehat{n})E(\widehat{c}) + E(|\hat{n}|^2\widetilde{n})\overline{E(\widehat{c})} - E(|\hat{n}|^2)E(\widehat{c})\overline{E(\hat{n})} - E(|\hat{n}|^2)\overline{E(\widehat{c})}E(\hat{n}) = 0$ since $E(|\hat{n}|^2\widehat{n}) = E(\hat{n}) = 0$.
- $\text{var}(\widehat{c\hat{n}} + \widetilde{c\hat{n}}) = E((\widehat{c\hat{n}} + \widetilde{c\hat{n}})^2) = 2\text{Re}(E(\widehat{c^2})\overline{E(\widehat{n^2})}) + 2E(|\hat{c}|^2)E(|\hat{n}|^2) \leq 4E(|\hat{c}|^2)E(|\hat{n}|^2)$ since for any complex number z , $\text{Re}(z) \leq |z|$, and for any random variable V , $|E(V)| \leq E(|V|)$.
- concerning $\text{var}(|\hat{n}|^2) = E(|\hat{n}|^4) - E(|\hat{n}|^2)^2$. By definition of \hat{n} ,

$$E(|\hat{n}|^4(\boldsymbol{\xi})) = \sum E(n_{x,y}n_{x',y'}n_{x'',y''}n_{x''',y'''})e^{-2i\pi\left(\frac{x-x'+x''-x'''}{L}\xi + \frac{y-y'+y''-y'''}{L}\eta\right)} \quad (5)$$

where the summation is over (x, y) , (x', y') , (x'', y'') , and (x''', y''') spanning the image domain. Now, $E(n_{x,y}n_{x',y'}n_{x'',y''}n_{x''',y'''}) = \sigma^4$ if $(x, y, x'', y'') = (x', y', x''', y''')$, or if $(x, y, x', y') = (x'', y'', x''', y''')$, or if $(x, y, x', y') = (x''', y''', x'', y'')$, and it simplifies into $E(n_{x,y}n_{x',y'}n_{x'',y''}n_{x''',y'''}) = 0$ otherwise. Thus,

$$E(|\hat{n}(\boldsymbol{\xi})|^4) = \sum_{\mathbf{x}, \mathbf{x}'} \left(2 + e^{-4i\pi\langle \mathbf{x} - \mathbf{x}', \boldsymbol{\xi} \rangle}\right) \sigma^2(\mathbf{x})\sigma^2(\mathbf{x}') \quad (6)$$

$$= 2\sigma^4 + \left| \sum_{\mathbf{x}} \sigma^2(\mathbf{x})e^{-4i\pi\langle \mathbf{x}, \boldsymbol{\xi} \rangle} \right|^2 \quad (7)$$

Since $\text{var}(|\hat{n}|^2) = E(|\hat{n}|^4) - \sigma^4$,

$$\text{var}(|\hat{n}|^2(\boldsymbol{\xi})) = \sigma^4 + \left| \widehat{\sigma^2}(\boldsymbol{\xi}) \right|^2 \leq 2\sigma^4 \quad (8)$$

the latter inequation holding thanks to Bessel inequality.

To sum up the discussion, we can conclude that:

$$\text{var}(|\widehat{p_0}|^2) = \text{var}(|\widehat{c}|^2) + \sigma^4 + V \quad (9)$$

where $V \leq 4\sigma^2 E(|\widehat{c}|^2) + \sigma^4$.

2.2 Estimating the statistics of a non-textured random patch

In Section 3, texture components are detected in the power spectrum of a patch as deviations to a Gaussian model of the power spectrum of non-textured random patches. We first need to empirically estimate the expectation and variance of the power spectrum of a non-textured patch p_0 . We can see from Equation 3 and 9 that this requires to estimate the expectation and variance of the power spectrum of the cartoon component c , and the noise total variance $\sigma^2 = \sum_{\mathbf{x}} \sigma^2(\mathbf{x})$.

2.2.1 Estimating cartoon statistics

Although the approach of [12] gives a coarse cartoon output which is often quite smooth, it is fast to compute. We therefore take it as an initial guess of the cartoon component of any patch in the image. Smoothness is even an advantage here. Indeed, it makes it possible to assume that noise has been removed from the cartoon component. In order to estimate both the expectation $E(|\hat{c}|^2)$ and the variance $\text{var}(|\hat{c}|^2)$ for any random patch $p = c + n$, we make use of a non-local estimation approach. More precisely, for a patch $p_{\mathbf{x}}$ centered at any pixel \mathbf{x} , we extract from the input image the set \mathcal{S} of patches such that their power spectrum are the N nearest neighbors of $|\hat{p}_{\mathbf{x}}|^2$ in the sense of the Euclidean norm. These patches are thus likely to share the same texture and to be realizations of the same random process. Weighted sample estimates of the expectation $E(|\hat{c}|^2)$ and variance $\text{var}(|\hat{c}|^2)$ are calculated based on the cartoon components of the patches in \mathcal{S} . The weights take into account dissimilarities and are proportional to $e^{-||\hat{c}|^2 - |\hat{s}|^2||/\lambda(\mathbf{x})}$ for any patch $s \in \mathcal{S}$, $\lambda(\mathbf{x})$ being a normalizing parameter defined later.

In practice, the proposed procedure is made of two steps. First, the power spectra of image patches and of the corresponding cartoon components are extracted from the whole image. In order to deal with shape edges and avoid mixing texture information from two adjacent regions, the contribution of each pixel is weighted as a non-decreasing function of the similarity to the central pixel in the cartoon domain. Indeed, it has been shown in several papers [36, 37] that adapting patches to the shape is a good way to keep non-local methods from mixing information from overlapping shapes. Moreover, in order to deal with border discontinuities and avoid spectral leakage, we also multiply patches by a Gaussian window. Therefore, denoting by I the input image, by $\mathcal{D} = \{1 \dots X\} \times \{1 \dots Y\}$ its domain, and by C^c its coarse cartoon representation obtained by [12], we actually compute modified windowed Fourier transforms defined at each pixel \mathbf{x} as:

$$\mathcal{I}_{\mathbf{x}}(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \mathcal{D}} g_{\alpha}(\mathbf{x} - \mathbf{y}) g_{\beta}(C^c(\mathbf{x}) - C^c(\mathbf{y})) I(\mathbf{y}) e^{-2i\pi \langle \mathbf{y}, \boldsymbol{\xi} \rangle} \quad (10)$$

$$\mathcal{C}_{\mathbf{x}}(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \mathcal{D}} g_{\alpha}(\mathbf{x} - \mathbf{y}) g_{\beta}(C^c(\mathbf{x}) - C^c(\mathbf{y})) C^c(\mathbf{y}) e^{-2i\pi \langle \mathbf{y}, \boldsymbol{\xi} \rangle} \quad (11)$$

where g_{α} and g_{β} are Gaussian functions of standard deviation α and β , g_{α} being truncated to define a $L \times L$ patch. The corresponding power spectra are denoted by $\mathcal{P}_{\mathbf{x}}^I$ and $\mathcal{P}_{\mathbf{x}}^C$. By definition, $\mathcal{P}_{\mathbf{x}}^I(\boldsymbol{\xi}) = |\mathcal{I}_{\mathbf{x}}(\boldsymbol{\xi})|^2$ and $\mathcal{P}_{\mathbf{x}}^C(\boldsymbol{\xi}) = |\mathcal{C}_{\mathbf{x}}(\boldsymbol{\xi})|^2$ for any frequency $\boldsymbol{\xi} = (\xi, \eta) \in \{1, \dots, L\} \times \{1, \dots, L\}$.

We extract these power spectra in such a way that their overlap is limited in order to avoid bias when estimating sample mean and sample variance. In practice, we impose that the patch center \mathbf{x} lies on a subsampled grid \mathcal{D}_s made of regularly distributed pixel locations at a distance of s pixels.

The second step consists in estimating at any pixel $\mathbf{x} \in \mathcal{D}$ the expectation and variance of the power spectrum of cartoon components based on the $\mathcal{P}_{\mathbf{x}_i}^I$ and $\mathcal{P}_{\mathbf{x}_i}^C$ calculated above for $\mathbf{x}_i \in \mathcal{D}_s$. At any $\mathbf{x} \in \mathcal{D}$, these statistics are estimated over the set of the N cartoon power spectra

$$\{\mathcal{P}_{\mathbf{x}_i}^C, \mathbf{x}_i \in \mathcal{D}_s, 1 \leq i \leq N\}, \quad (12)$$

with the \mathbf{x}_i defined such that

$$\{\mathcal{P}_{\mathbf{x}_i}^I, \mathbf{x}_i \in \mathcal{D}_s, 1 \leq i \leq N\} \quad (13)$$

is the set of the N -nearest neighbors of the query power spectrum $\mathcal{P}_{\mathbf{x}}^I$. We use weighted

estimates to take the dissimilarity of the nearest power spectra into account. For any ξ ,

$$\tilde{E}(\mathcal{P}_x^C(\xi)) = \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i \mathcal{P}_{x_i}^C(\xi) \quad (14)$$

$$\widetilde{\text{var}}(\mathcal{P}_x^C(\xi)) = \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i \left(\mathcal{P}_{x_i}^C(\xi) - \tilde{E}(\mathcal{P}_x^C(\xi)) \right)^2 \quad (15)$$

where the weights w_i satisfy

$$w_i = e^{-\|\mathcal{P}_x^I - \mathcal{P}_{x_i}^I\|^2 / \lambda(\mathbf{x})^2} \quad (16)$$

$\|\cdot\|$ being the Euclidean norm of the components of frequency larger than $2/L$ cycles per pixel (to remove the effect of very low frequency changes), and $\lambda(\mathbf{x})$ being a normalization parameter equal to the median of the distances of the N nearest neighbors of \mathcal{P}_x^I . The goal of this normalization, similarly to the so-called contextual dissimilarity measure [38], is to adapt the weights to the non-homogeneous distribution of the power spectra.

The nearest neighbors of the current power spectrum are found over the whole image by using the efficient FLANN library [39].

It should be noted that, patch being here the result of the pixelwise multiplication of the underlying image and an analysis window, it is necessary to assume the noise amplitude to vary from pixel to pixel even if a Gaussian white noise affects the image to be decomposed.

2.2.2 Estimating noise variance

We see from Equation 3 that the noise total variance σ^2 can be estimated as the empirical mean of $|\hat{p}_0(\xi)|^2 - |\hat{c}(\xi)|^2$ for any ξ . The non-textured patch p_0 is unknown, and only a coarse estimation of c is available. As a consequence, for any ξ , we assume that $\sigma^2 \simeq \tilde{E}(\mathcal{P}_x^I(\xi)) - \tilde{E}(\mathcal{P}_x^C(\xi))$. This equation mainly differs from the preceding one in that the power spectrum \mathcal{P}_x^I is affected by texture components. Since textures have a sparse Fourier representation, these components only affect a few Fourier components, thus averaging the power spectrum difference over a range of frequencies smoothes them out. Moreover, we average over high frequencies since low and medium frequencies are expected to have residual image content not purely made of noise, cf. [40, 41]. Consequently, we estimate σ^2 as:

$$\tilde{\sigma}^2 = \frac{1}{M} \sum_{\|\xi\| > H} \tilde{E}(\mathcal{P}_x^I(\xi)) - \tilde{E}(\mathcal{P}_x^C(\xi)) \quad (17)$$

where H is a frequency threshold and M is the number of discrete frequencies above H . In practice, we take $H = 0.5$ cycle per pixel.

2.2.3 Estimating statistics of the power spectrum of a noisy non-textured patch

Based on Equation 3, we eventually estimate the expectation of the power spectrum \mathcal{P}_x of a non-textured noisy patch. Concerning the variance, we use the upper bound given by Equation 9. Consequently,

$$\tilde{E}(\mathcal{P}_x(\xi)) = \tilde{E}(\mathcal{P}_x^C(\xi)) + \tilde{\sigma}^2 \quad (18)$$

$$\widetilde{\text{var}}(\mathcal{P}_x(\xi)) = \widetilde{\text{var}}(\mathcal{P}_x^C(\xi)) + 2\tilde{\sigma}^4 + 4\tilde{\sigma}^2 \tilde{E}(\mathcal{P}_x(\xi)) \quad (19)$$

3 A non-local dual-domain cartoon and texture decomposition

The preceding section gives the expectation and variance of the power spectrum components of non-textured noisy patches, empirically estimated with a tentative coarse cartoon and texture decomposition (Equations 18 and 19). We are therefore able to detect texture components as deviations to this model, since these components have a larger amplitude than expected under a null hypothesis assuming a non-textured patch.

3.1 Statistical hypothesis testing

Given an image patch p , a significant spectrum component is detected at frequency ξ provided that the p -value $\Pr_{\mathcal{H}_0}(|\hat{p}(\xi)|^2 \geq |\hat{p}_0(\xi)|^2)$ is below some level of confidence α . The null-hypothesis \mathcal{H}_0 is that of a non-textured random patch p_0 . Under a Gaussian distribution assumption, a texture component is therefore detected at ξ as soon as:

$$G\left(\frac{\mathcal{P}_x^I(\xi) - \tilde{E}(\mathcal{P}_x(\xi))}{\sqrt{\widetilde{\text{var}}(\mathcal{P}_x(\xi))}}\right) \leq \alpha \quad (20)$$

where \mathcal{P}_x^I is the power spectrum of the image patch centered at \mathbf{x} , $\tilde{E}(\mathcal{P}_x(\xi))$ and $\widetilde{\text{var}}(\mathcal{P}_x(\xi))$ are given by Equations 18 and 19, respectively, and G is the tail distribution function (complementary cumulative distribution function) of the standard normal distribution. Here, the estimation of the variance being an upper bound of the actual variance, some significant detections may be missed.

Since a statistical test is performed for any frequency, we use the Bonferroni correction [42] to control the false alarm over each patch of size $L \times L$ under ε , thus:

$$\alpha = \frac{\varepsilon}{L^2} \quad (21)$$

where ε is the familywise error-rate, set here equal to 5%.

We do not take into account the correlation matrix of the power spectrum components, since estimating the correlations would require a large number of samples, which is out of reach here. It should be noted that a recent paper addresses the estimation of the correlation matrix of patches [30].

3.2 Cartoon and texture components

Section 3.1 gives a framework to detect texture components in the spectrum of any patch. For any patch centered at pixel \mathbf{x} , we thus define a very simple bandpass filter M_x : $M_x(\xi) = 1$ if a significant texture component is detected at frequency ξ , and $M_x(\xi) = 0$ otherwise. This gives a non-linear bandpass filter through a windowed Fourier transform of the input image I . If $\hat{I}(\mathbf{x}, \xi)$ denotes the windowed Fourier transform of I , such that:

$$\hat{I}(\mathbf{x}, \xi) = \sum_{\mathbf{y} \in \mathcal{D}} I(\mathbf{x}) g_\alpha(\mathbf{x} - \mathbf{y}) e^{-2i\pi \langle \mathbf{x}, \xi \rangle} \quad (22)$$

then the texture component T is retrieved as the inverse windowed Fourier transform of the product $M_x(\xi) \hat{I}(\mathbf{x}, \xi)$. The cartoon component C is simply such that $I = C + T$.

Data: An input image I , domain $\mathcal{D} = \{1 \dots X\} \times \{1 \dots Y\}$.

Parameters: σ_{CT} , α , β , N .

- 1 Compute a tentative coarse cartoon representation (denoted by C^c) from I with [12], scale-factor parameter σ_{CT} ;
 - 2 **for** any pixel $\mathbf{x} = (x, y) \in \mathcal{D}_s$ **do**
 - 3 Calculate the modified windowed Fourier transforms for any frequency $\boldsymbol{\xi}$:
 - 4 $\mathcal{I}_x(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \mathcal{D}} g_\alpha(\mathbf{x} - \mathbf{y}) g_\beta(C^c(\mathbf{x}) - C^c(\mathbf{y})) I(\mathbf{y}) e^{-2i\pi \langle \mathbf{y}, \boldsymbol{\xi} \rangle}$
 - 5 $\mathcal{C}_x(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \mathcal{D}} g_\alpha(\mathbf{x} - \mathbf{y}) g_\beta(C^c(\mathbf{x}) - C^c(\mathbf{y})) C^c(\mathbf{y}) e^{-2i\pi \langle \mathbf{y}, \boldsymbol{\xi} \rangle}$
 - 6 and the corresponding power spectra $\mathcal{P}_x^I(\boldsymbol{\xi}) = |\mathcal{I}_x(\boldsymbol{\xi})|^2$ and $\mathcal{P}_x^C(\boldsymbol{\xi}) = |\mathcal{C}_x(\boldsymbol{\xi})|^2$.
 - 7 **end**
 - 8 **for** any pixel $\mathbf{x} \in \mathcal{D}$ **do**
 - 9 Find the N nearest image-patch power spectra $\mathcal{P}_{\mathbf{x}_1}^I, \dots, \mathcal{P}_{\mathbf{x}_N}^I$ to \mathcal{P}_x^I with the FLANN library [39];
 - 10 Calculate sample mean $\tilde{E}(\mathcal{P}_x^C(\boldsymbol{\xi}))$ (Eq. 14) and sample variance $\widetilde{\text{var}}(\mathcal{P}_x^C(\boldsymbol{\xi}))$ (Eq. 15) based on the \mathbf{x}_i ;
 - 11 Calculate estimation of total variance $\tilde{\sigma}^2$ (Eq. 17);
 - 12 Calculate estimations of expectation $\tilde{E}(\mathcal{P}_x(\boldsymbol{\xi}))$ (Eq. 18) and variance $\widetilde{\text{var}}(\mathcal{P}_x(\boldsymbol{\xi}))$ (Eq. 19) of a non-textured patch;
 - 13 Detect texture components at $\boldsymbol{\xi}$ through statistical hypothesis testing:
 - 14
$$M_x(\boldsymbol{\xi}) = \begin{cases} 1 & \text{if } G\left(\frac{\mathcal{P}_x^I(\boldsymbol{\xi}) - \tilde{E}(\mathcal{P}_x(\boldsymbol{\xi}))}{\sqrt{\widetilde{\text{var}}(\mathcal{P}_x(\boldsymbol{\xi}))}}\right) \leq 0.05/L^2 \\ 0 & \text{otherwise} \end{cases}$$
 - 15 **end**
 - 16 Apply the bandpass filter $M_x(\boldsymbol{\xi})$ to the windowed Fourier transform of I to retrieve texture T and cartoon $C = I - T$, i.e., $T(\mathbf{x}) = \frac{1}{L^2} \sum_{\mathbf{y} \in \mathcal{D}} \sum_{\boldsymbol{\xi} \in \mathcal{D}} \hat{T}(\mathbf{y}, \boldsymbol{\xi}) e^{2i\pi \langle \mathbf{x}, \boldsymbol{\xi} \rangle}$ where $\hat{T}(\mathbf{x}, \boldsymbol{\xi}) = M_x(\boldsymbol{\xi}) \sum_{\mathbf{y} \in \{1 \dots X\} \times \{1 \dots Y\}} g_\alpha(\mathbf{x} - \mathbf{y}) I(\mathbf{y}) e^{-2i\pi \langle \mathbf{y}, \boldsymbol{\xi} \rangle}$.
- Result:** A cartoon component C and a texture component T .

Algorithm 1: Proposed algorithm for non-local dual-domain cartoon and texture decomposition.

3.3 Algorithm and parameters

Algorithm 1 summarizes the proposed non-local dual-domain cartoon and texture composition. Parameter σ_{CT} in [12] to obtain the tentative coarse cartoon is set so that textures are discarded from the cartoon but is small enough to prevent discarding high frequency details. In Section 4, $\sigma_{CT} = 2$ unless otherwise specified. Patch width L and parameter α satisfy $\alpha = L/5$; their value define the trade-off between shape and texture. Any pattern showing a low frequency with respect to $1/\alpha$ will not be detected as a texture, basically because it does not give a spike in patch power spectra. A trade-off has to be made between the ability to detect low-period patterns (which requires large L) and the need to have a large enough sample of non-overlapping patch for statistical estimation (which bounds L). We extract patches on a subsampling grid \mathcal{D}_s of step $s = L/4$. Concerning the experiments of Section 4, a typical value for L will be $L = 32$ pixels, which permits detecting texture patterns of frequency larger than $1/16 \text{ pixel}^{-1}$. Note that such a patch size is much larger than typical values in patch methods dedicated to noise removal. Parameter β prevents mixing texture information with shapes with a gray value

larger or smaller than the current shape. Because a Gaussian function is used, a difference of the cartoon components larger than 2β indeed multiply the corresponding gray value by e^{-2} . The value of β obviously depends on the contrast of the image, low-contrasted images indeed require smaller values. We set $\beta = 20$ in Section 4 unless otherwise stated. The number of nearest neighbors N is set to 20.

4 Experimental results

This section presents some experiments. We deal with color images by designing the bandpass filter of Section 3.2 in the luminance channel, the same filter being eventually used in the RGB channels. The texture components shown in this section actually correspond to the 8-bit image $128 + 3T$ so that they appear much salient in the figures. The reader is asked to zoom in on the images in the electronic version of the paper.

Since the cartoon and texture decomposition problem is based on perceptual clues, no ground truth is available. Overall, we would like the cartoon edges to be preserved and the stationary patterns to be in the texture part. As mentioned in [12], the distinction between texture and cartoon is not very clear because at a certain scale a texture element becomes a geometric shape. We visually compare the proposed algorithm with two state-of-the-art algorithms, namely the output of the block nuclear norm based algorithm [11], which is claimed to outperform variational methods such as [10], and the directional non-linear filter proposed in [14], which certainly achieves the best trade-off between the quality of the cartoon / texture separation and computation time. We use the software implementation kindly provided by the authors of [11] and [15]. Note that the implementation of [11] only gives gray-level outputs. The default parameters of [11] are used, and the scale-factor parameter σ_{CT} in [15] is adjusted to obtain the best visual results. Our algorithm is implemented as a publicly available Matlab code.

To begin with, we illustrate the proposed algorithm with a famous image containing several different textures. Figure 1 shows the original image, the cartoon component given by the fast non-linear filter of [12] (this is the coarse decomposition used in our algorithm) and the output of the proposed approach. As we can see, while the texture component has mostly been taken out, the cartoon component of [12] is quite blurry. Comparing with the output of [11] and [15], we can see that the output of the proposed algorithm tends to better separate texture and cartoon, with less blurry shape edges and a larger part of the texture component well separated from the cartoon as, for instance, in the tablecloth, in the chair on the background, and even in narrow shapes as in the folds of the scarf. Most edges cannot be seen in the texture part, except for periodic patterns on the face for instance. It is a positive feature of the proposed algorithm, which means that high frequency components corresponding to edges are correctly detected as non-significant spectrum components. Here, the scale-factor parameter σ_{CT} of [15] is difficult to set: $\sigma_{CT} = 2$ keeps textures in the cartoon part, and $\sigma_{CT} = 3$ (not shown here, see the companion document) gives a quite smooth cartoon with edge information in the texture.

Figure 2 illustrates the behavior of the algorithm for several patches. We can see that similar patches (in the sense of power spectrum similarity) are weighted according to the texture similarity. The numerical values indicated here are the $w_i / \sum w_i$ of Equation 16; they would be equal to $1/N = 0.05$ if the patches were evenly distributed. Moreover, we can see that the expected cartoon power spectrum actually varies from patch to patch, which illustrates that they come from different Gaussian processes. The hypothesis testing framework allows obtaining the localization of the texture spikes in the Fourier domain, and detects no significant texture components if no textures are present, as in the third patch.

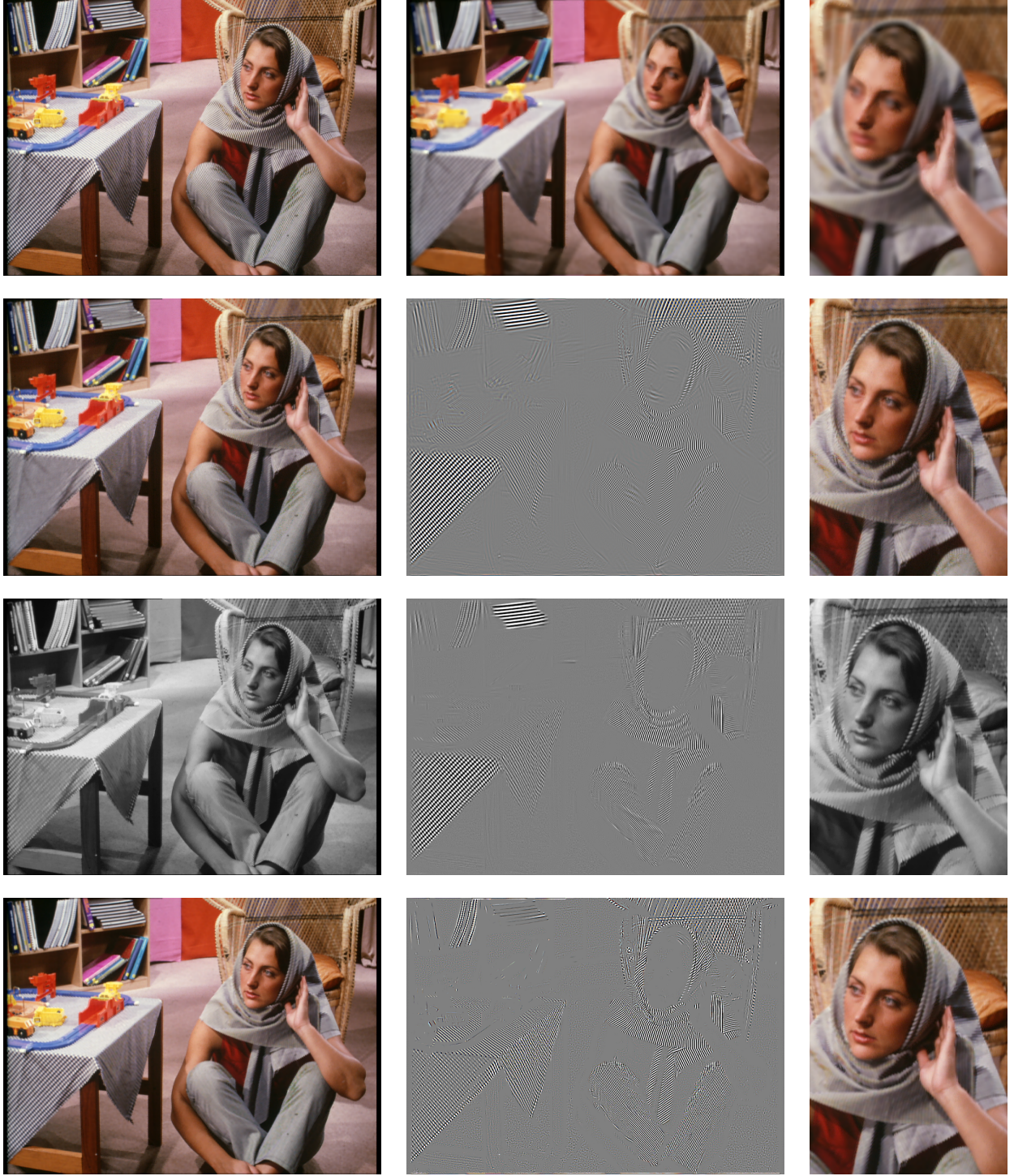


Figure 1: Barbara. First row: original image and output of [12]. Second row: output of the proposed algorithm. Third row: output of [11]. Fourth row: output of [15] ($\sigma_{CT} = 2$). From the left to the right: cartoon, texture (except for first row), and close-up.

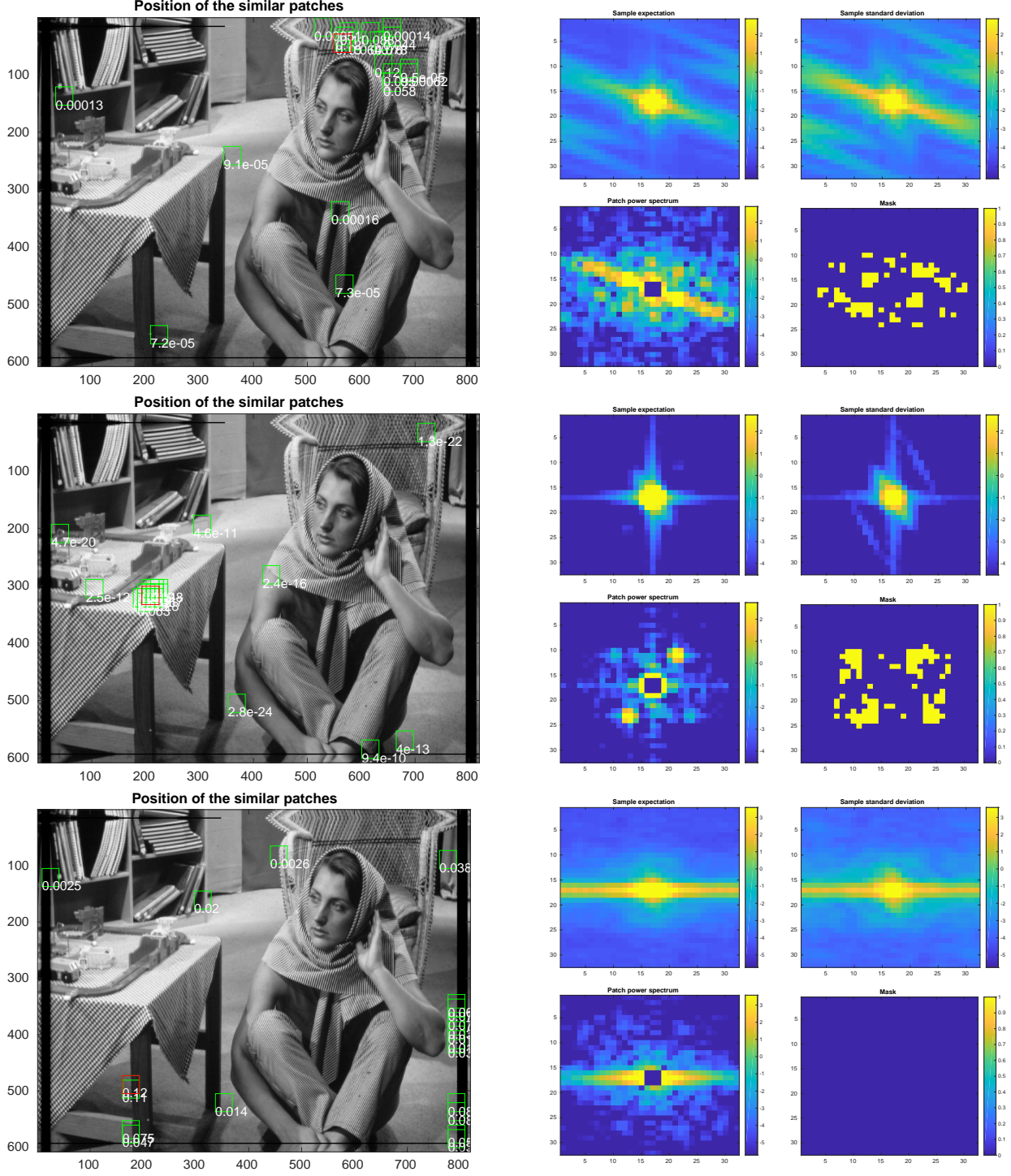


Figure 3 is another experiment with textured building facades¹. Compared to the other methods, our cartoon component appears more realistic, probably because the noise component is kept in it, while the other methods minimize the total variation of the cartoon part, either explicitly or implicitly, thus smoothing out the noise.

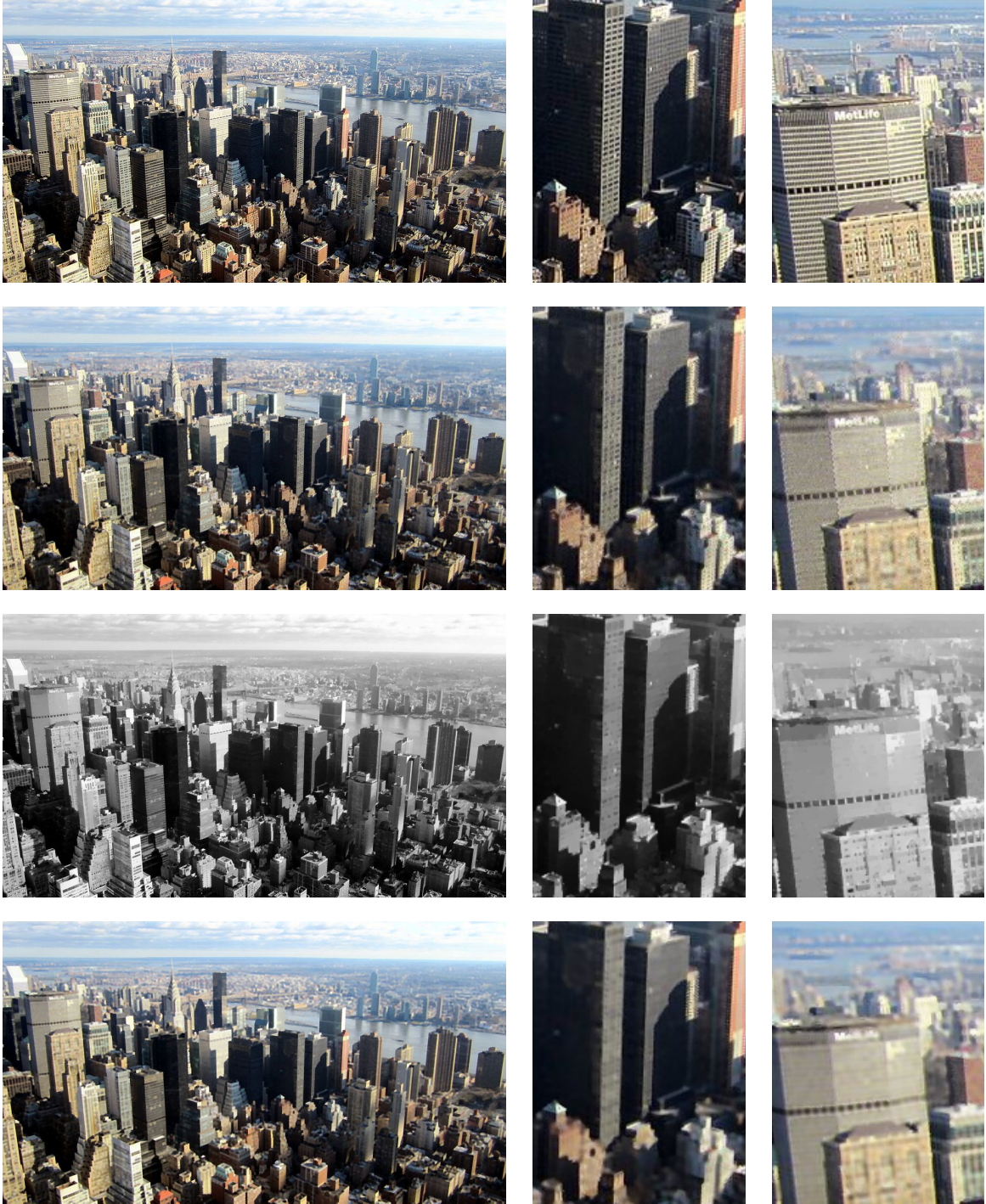


Figure 3: Manhattan. From top to bottom: original image, output of the proposed algorithm, output of [11], output of [15].

¹Image credit: By Hakilon - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=18111836>

Figure 4² also illustrates that our method retrieves texture details that other methods miss, such as tiles on roofs.



Figure 4: Kodim08. Top left: original image. Top right: output of the proposed algorithm. Bottom left: output of [11]. Bottom right: output of [15].

Figure 5 shows a detail of Norman Rockwell's painting, *Our Heritage*³. It is an application of cartoon and texture decomposition to canvas pattern removal. The canvas of the paintings are sometimes visible and need to be removed for further processing, as explained in [29]. Although this problem has been the subject of works from the eighties [43], it is still an open question [29, 44]. It can be seen as a cartoon and texture problem as the canvas is locally not visible if the painting layer is too thick. Here, σ_{CT} has been set to 3 for obtaining the tentative coarse cartoon with [12], and $\beta = 30$ since the image is quite contrasted. As we can see, the proposed algorithm is interesting in this application domain since it removes the quasi-periodic component of the canvas pattern (contrary to [11] and in a better way than [12]), and it still keeps sharp edges.

²Image source: Kodak Image Suite. <http://r0k.us/graphics/kodak/>

³Image source: <http://muddycolors.blogspot.fr/2013/10/norman-rockwell-american-originals.html>

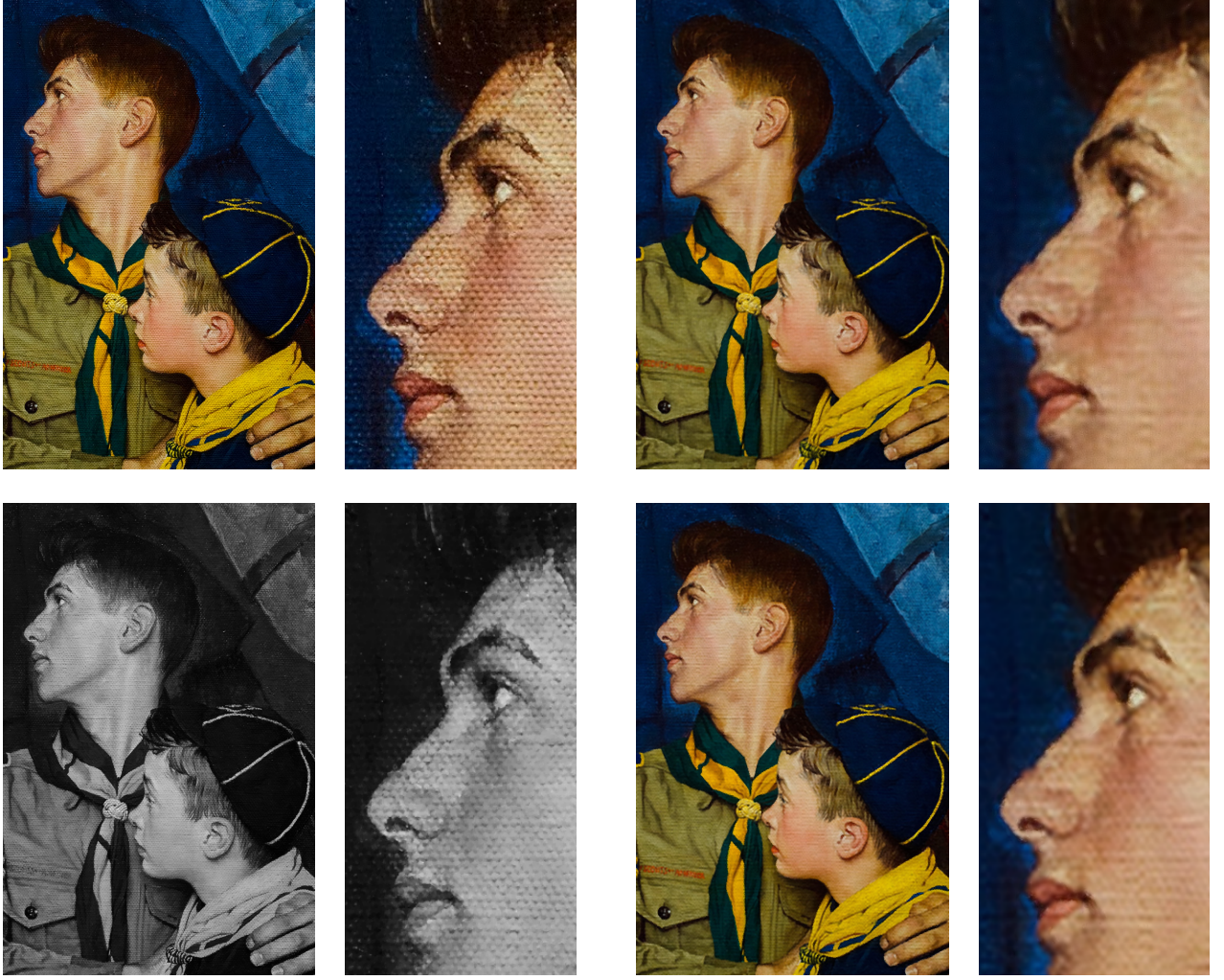


Figure 5: Rockwell. Top left: original image. Top right: output of the proposed algorithm. Bottom left: output of [11]. Bottom right: output of [15].

Figure 6 shows an application to remote sensing image processing⁴. Remote sensing image processing dates back to the dawn of robotic probe imaging [46]. Such images are often impaired by electrical interferences during image acquisition, miscalibrated sensors, or missing data, causing line dropout, striping, banding, or more complex background noise. In this image, the size of the patches L has been set to 64, since we want to capture texture information such as the checkerboard noise pattern which has a periodicity of approximately 20 pixels. The contrast being low, β is set to 10. As we can see, the proposed algorithm removes most of the background noise (contrary to [15]) while keeping a quite sharp output (in contrast to [11]).

As explained in Section 1, assessing the separation between cartoon and texture is difficult since it depends on perceptual clues. Although such an experiment may look somewhat contrived, it is possible to test the algorithms discussed here on synthetic data. Figure 7 shows cartoon and texture decompositions of a synthetic image, built by adding to a piecewise constant 8-bit image⁵ a texture image whose quadrants are made of three sine-wave textures, the fourth quadrant being kept untextured. As we can see, the outputs of [11] and [15] show somewhat mixed cartoon and texture, cartoon edges being noticeable in the retrieved texture. This

⁴Image source: <http://www.imageprocessingplace.com/> [45], see also [46]

⁵Image source: <http://sipi.usc.edu/database/>

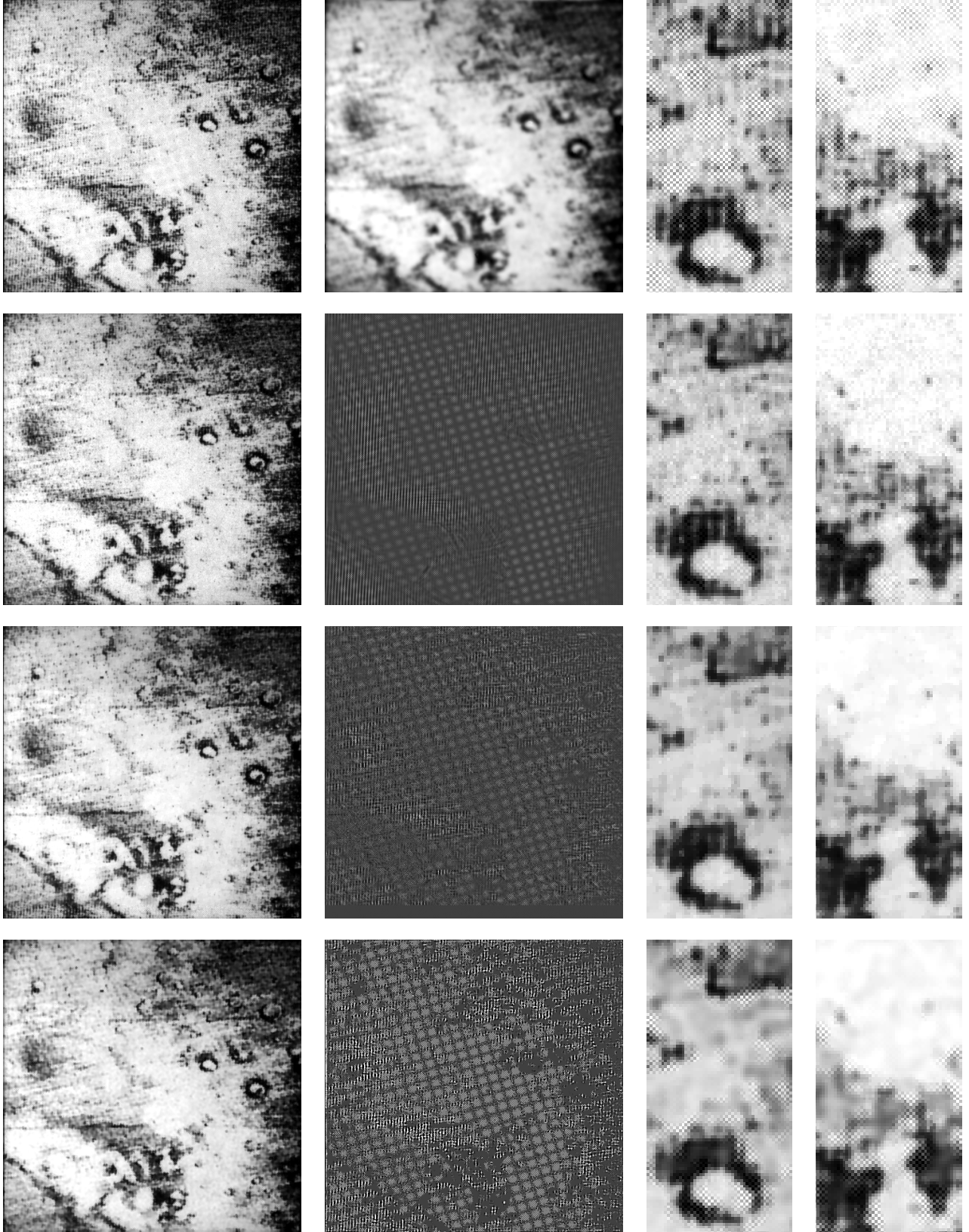


Figure 6: Mariner6. The contrast is augmented for a better visibility. First row: original image, output of [12], details of the original image with the “checkerboard” noise clearly visible. Second row: output of the proposed algorithm. Third row: output of [11]. Fourth row: output of [15]. Second, third, and fourth row, from left to right: cartoon, texture, and close-up view of the cartoon component.

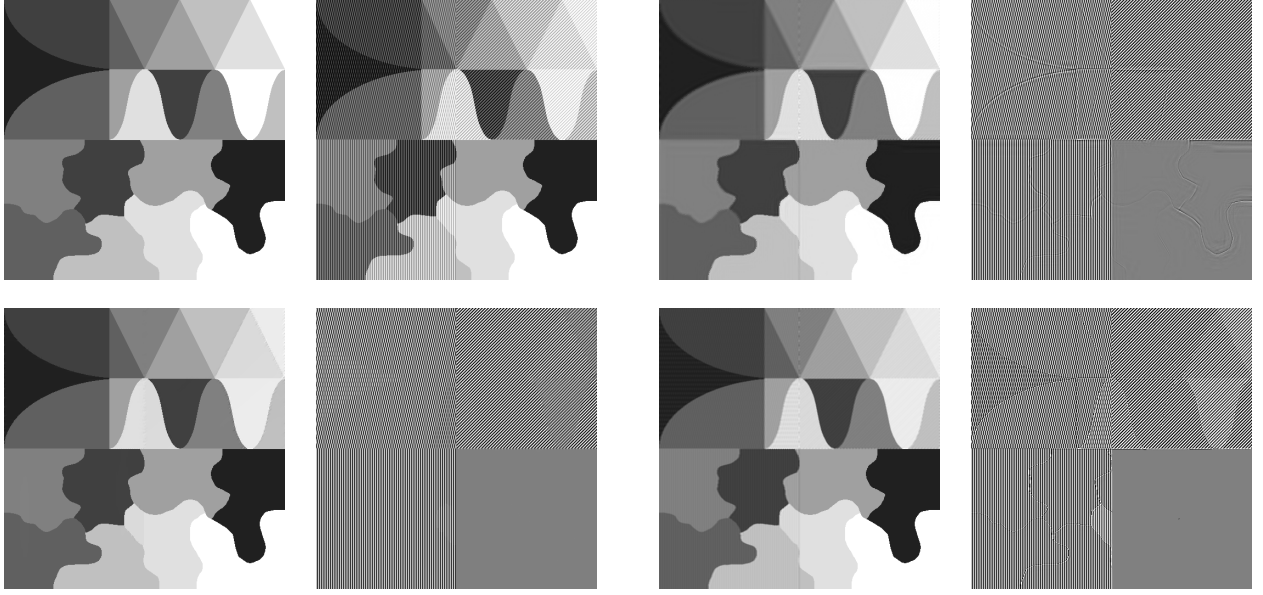


Figure 7: **Synthetic.** First row: ground truth cartoon component, input image, cartoon retrieved by the proposed algorithm, texture retrieved by the proposed algorithm. Second row: cartoon retrieved by [11], texture retrieved by [11], cartoon retrieved by [14], texture retrieved by [14].

phenomenon is less visible with the proposed algorithm. Here, the proposed algorithm and the one of [11] are used with their default parameters, and $\sigma_{CT} = 2$ in [15]. Table 1 gives a quantitative assessment with the root-mean-square error (RMSE) between the retrieved cartoon (resp. texture) component and ground-truth cartoon (resp. texture) component. The RMSE criterion confirms the visual impression (the lower the values, the better the decomposition). Although the cartoon component of [11] looks sharper, some parts have a grayer aspect than in the original image, which explains the larger RMSE. With the proposed algorithm, RMSE are equal for cartoon and texture components because the sum of both components exactly matches the input image. Texture normalization used in the software program [15] may explain why RMSE of cartoon and texture differ for this method in this particular experiment.

Table 1: RMSE with synthetic data

	proposed method	[11]	[15]
cartoon	2.72	4.51	6.15
texture	2.72	6.04	11.42

To conclude this discussion, Table 2 gives computation times on a desktop equipped with a 12-core Intel Xeon E5-2650 v4 @ 2.20GHz CPU and 64 Gb memory. We can see that the proposed method, implemented in a non-optimized Matlab software, is much faster than the one of [11]. In all cases, the computation time of the preprocessing step [12] in the proposed algorithm is less than 0.3 seconds.

Table 2: Computation times in seconds

Image	Size (in pixels)	proposed method	[11]	[15]
Barbara (Fig. 1)	567×787	35.5	332.1	6.5
Manhattan (Fig. 3)	575×1024	43.7	518.9	9.2
Kodim08 (Fig. 4)	512×768	29.7	375.5	6.3
Rockwell (Fig. 5)	1600×1060	124.3	967.9	47.8
Mariner6 (Fig. 6)	461×471	15.1	168.2	1.2
Synthetic (Fig. 7)	512×512	20.8	172.3	3.4

5 Conclusion

This paper presents a novel patch-based approach to cartoon and texture decomposition. Fourier components of textures are detected in patches through statistical hypothesis testing, the null model being estimated in a non-local way over a tentative coarse cartoon representation. The proposed algorithm achieves a good compromise between computation time and accuracy.

Acknowledgment

The authors of [11], [12] and [14] are gratefully acknowledged for making their software publicly available.

References

- [1] C. Knaus and M. Zwicker. Dual-domain filtering. *SIAM Journal on Imaging Sciences*, 8(3):1396–1420, 2015.
- [2] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):490–530, 2005.
- [3] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, 1992.
- [4] T.F. Chan and S. Esedoglu. Aspects of total variation regularized L1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

- [5] J. Gilles and Y. Meyer. Properties of $BV - G$ structures + textures decomposition models. application to road detection in satellite images. *IEEE Transactions on Image Processing*, 19(11):2793–2800, 2010.
- [6] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Society, 2001.
- [7] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition – modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.
- [8] J.-L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, 2005.
- [9] J. Fehrenbach, P. Weiss, and C. Lorenzo. Variational algorithms to remove stationary noise: Applications to microscopy imaging. *IEEE Transactions on Image Processing*, 21(10):4420–4430, 2012.
- [10] H. Schaeffer and S. Osher. A low patch-rank interpretation of texture. *SIAM Journal on Imaging Sciences*, 6(1):226–262, 2013.
- [11] S. Ono, T. Miyata, and I. Yamada. Cartoon-texture image decomposition using blockwise low-rank texture characterization. *IEEE Transactions on Image Processing*, 23(3):1128–1142, 2014.
- [12] A. Buades, T. M. Le, J.-M. Morel, and L. A. Vese. Fast cartoon + texture image filters. *IEEE Transactions on Image Processing*, 19(8):1978–1986, 2010.
- [13] A. Buades, T. Le, J.-M. Morel, and L. Vese. Cartoon+texture image decomposition. *Image Processing On Line*, 1:1–8, 2011.
- [14] A. Buades and J.L. Lisani. Directional filters for color cartoon+texture image and video decomposition. *Journal of Mathematical Imaging and Vision*, 55(1):125–135, 2016.
- [15] A. Buades and J.-L. Lisani. Directional filters for cartoon + texture image decomposition. *Image Processing On Line*, 6:75–88, 2016.
- [16] B. Ham, M. Cho, and J. Ponce. Robust guided image filtering using nonconvex potentials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):192–207, 2018.
- [17] Y. Sun, S. Schaefer, and W. Wang. Image structure retrieval via L_0 minimization. *Transactions on Visualization and Computer Graphics*, 24(7):2129–2139, 2018.
- [18] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume LNCS 8691, pages 815–830, Zurich, Switzerland, 2014.
- [19] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013.
- [20] L. Wang and D.-C. He. Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910, 1990.

- [21] B. Galerne, Y. Gousseau, and J.-M. Morel. Random phase textures: Theory and synthesis. *IEEE Transactions on Image Processing*, 20(1):257–267, 2011.
- [22] J.J. van Wijk. Spot noise texture synthesis for data visualization. *ACM SIGGRAPH Computer Graphics*, 25(4):309–318, 1991.
- [23] I. Aizenberg and C. Butakoff. A windowed Gaussian notch filter for quasi-periodic noise removal. *Image and Vision Computing*, 26(10):1347–1353, 2008.
- [24] M. Cannon, A. Lehar, and F. Preston. Background pattern removal by power spectral filtering. *Applied Optics*, 22(6):777–779, 1983.
- [25] G.A.A. Hudhud and M.J. Turner. Digital removal of power frequency artifacts using a Fourier space median filter. *IEEE Signal Processing Letters*, 12(8):573–576, 2005.
- [26] F. Sur and M. Grédiac. Automated removal of quasiperiodic noise using frequency domain statistics. *IS&T / SPIE Journal of Electronic Imaging*, 24(1):013003/1–19, 2015.
- [27] F. Sur. An a-contrario approach to quasi-periodic noise removal. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3841–3845, Québec City, Canada, 2015.
- [28] A. van der Schaaf and J.H. van Hateren. Modelling the power spectra of natural images: Statistics and information. *Vision Research*, 36(17):2759–2770, 1996.
- [29] B. Cornelis, H. Yang, A. Goodfriend, N. Ocon, J. Lu, and I. Daubechies. Removal of canvas patterns in digital acquisitions of paintings. *IEEE Transactions on Image Processing*, 26(1):160–171, 2017.
- [30] P. Arias and J.-M. Morel. Video denoising via empirical Bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision*, 60(1):70–93, 2018.
- [31] A. Houdard, C. Bouveyron, and J. Delon. High-dimensional mixture models for unsupervised image denoising (HDMI). Preprint HAL-01544249, 2017.
- [32] M. Lebrun, A. Buades, and J.-M. Morel. A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.
- [33] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2012.
- [34] J.P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3/4):419–426, 1961.
- [35] H. Liu, Y. Tang, and H.H. Zhang. A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central normal variables. *Computational Statistics and Data Analysis*, 53(4):853–856, 2009.
- [36] C.-A. Deledalle, V. Duval, and J. Salmon. Non-local methods with shape-adaptive patches (NLM-SAP). *Journal of Mathematical Imaging and Vision*, 43(2):103–120, 2012.
- [37] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing*, 16(5):1395–1411, 2007.

- [38] H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11, 2010.
- [39] M. Muja and D.G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.
- [40] N.N. Ponomarenko, V.V. Lukin, M.S. Zriakhov, A. Kaarna, and J. Astola. An automatic approach to lossy compression of AVIRIS images. In *Proceedings of the International Geoscience and Remote Sensing Symposium (IGRSS)*, pages 472–475, Barcelona, Spain, 2007.
- [41] M. Colom and A. Buades. Analysis and extension of the Ponomarenko et al. method, estimating a noise curve from a single image. *Image Processing On Line (IPOL)*, 3:173–197, 2013.
- [42] C.E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.
- [43] F. Heitz, H. Maitre, and C. de Couessin. Application of autoregressive models to fine arts painting analysis. *Signal Processing*, 13(1):1–14, 1987.
- [44] B. Cornelis, A. Doms, J. Cornelis, and P. Schelkens. Digital canvas removal in paintings. *Signal Processing*, 92(4):1166–1171, 2012.
- [45] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice-Hall, 3 edition, 2008.
- [46] T. C. Rindfleisch, J. A. Dunne, H. J. Frieden, W. D. Stromberg, and R. M. Ruiz. Digital processing of the Mariner 6 and 7 pictures. *Journal of Geophysical Research*, 76(2):394–417, 1971.