Stochastic super-resolution for Gaussian microtextures

Émile Pierret, Bruno Galerne

To cite this version:

Émile Pierret, Bruno Galerne. Stochastic super-resolution for Gaussian microtextures. SIAM Journal on Imaging Sciences, 2024, 18 (2), pp.1176-1207. ⟨10.1137/24M1657407⟩. ⟨hal-04560456v4⟩

HAL Id: hal-04560456

https://hal.science/hal-04560456v4

Submitted on 9 Jan 2025

Stochastic super-resolution for Gaussian microtextures *

Émile Pierret †   Bruno Galerne †‡

Abstract

Super-Resolution (SR) is the problem that consists in reconstructing images that have been degraded by a zoom-out operator. This is an ill-posed problem that does not have a unique solution, and numerical approaches rely on a prior on high-resolution images. While optimization-based methods are generally deterministic, with the rise of image generative models more and more interest has been given to stochastic SR, that is, sampling among all possible SR images associated with a given low-resolution input. In this paper, we construct an efficient, stable and provably exact sampler for the stochastic SR of Gaussian microtextures. Even though our approach is limited regarding the scope of images it encompasses, our algorithm is competitive with deep learning state-of-the-art methods both in terms of perceptual metric and execution time when applied to microtextures. The framework of Gaussian microtextures also allows us to rigorously discuss the limitations of various reconstruction metrics to evaluate the efficiency of SR routines. An implementation of our algorithm is available online.

Keywords: stochastic super-resolution, Gaussian textures, conditional simulation, kriging, super-resolution with a reference image

1 Introduction

Super-Resolution (SR) algorithms aim at producing a High Resolution (HR) image corresponding to a Low Resolution (LR) one, the main challenge being to restore sharp edges as well as high frequency texture content that are lost when applying the zoom-out operator. Since the set of HR images compatible with a given LR input is an affine subspace of high dimension, a strong prior on realistic images is necessary to recover HR images of high visual quality within this subspace. In recent works, this prior is generally conveyed either by exploiting a large datasets of HR images via deep learning models or using a reference HR image to specify an adapted model for the unknown HR image. The former solution is rather generic while the later is particularly adapted for the SR of texture images that benefits from a particular prior on local statistics of the HR unknown image. Besides, rather than being deterministic, several contributions propose to tackle the one to many dilemma via stochastic SR which consists in sampling among all acceptable HR images associated with an LR input. Motivated by such recent contributions, in this paper we solve the stochastic SR problem when the HR texture image is assumed to follow a stationary Gaussian distribution.

1.1 State of the art and related work

First deep learning approaches for SR propose to optimize the L_2 or L_1 reconstruction loss using Convolutional Neural Network (CNN) trained in a end-to-end fashion [14, 44, 38, 40]. However, the CNN outputs tend to be blurry as they correspond to the mean of plausible SR solutions [61]. To avoid this regression to the mean issue, other approaches rely on a perceptual loss [5, 64, 57], that is, an L_2 loss between pre-trained VGG features [58] introduced in [34], or Generative Adversarial Network (GAN) [26] conditioned on the LR input image [43, 65]. More generally SR is a linear inverse problem that can be tackled by most general deep learning techniques for inverse problems in imaging [51] that go beyond end-to-end learning, notably variational frameworks such as deep image prior [62, 63] and Plug-and-Play methods [35]. We do not explore this important body of literature in this work since we are more interested in stochastic methods.

---

[*] The authors acknowledge the support of the MISTIC project (ANR-19-CE40-005) and the CaSciModOT federation (Calcul Scientifique et Modélisation Orléans-Tours).

[†] Institut Denis Poisson – Université d'Orléans, Université de Tours, CNRS .(emile.pierret@univ-orleans.fr,`https://www.idpoisson.fr/pierret/`,bruno.galerne@univ-orleans.fr, `https://www.idpoisson.fr/galerne/`)/

[‡] Institut Universitaire de France (IUF) .

In contrast with the deterministic approaches mentioned above, stochastic SR algorithms sample within the set of plausible HR images consistent with the LR input. Lugmayr *et al.* [46] use a conditional normalizing flow [39] (a network with invertible layers) to learn this distribution via the change-of-variable formula. Diffusion or score-based models are generative networks which gradually learn to transform noise into samples of a distribution data [60, 29]. The diffusion framework can be used to train conditional diffusion models specific to SR [56]. Alternatively generic diffusion models can be used to provide approximate conditional sampling procedure for inverse problems such as SR [60, 37, 8, 56, 59], an approach that can be extended to latent diffusion models [55]. Alternatively, Jiang *et al.* [33] propose to wrap a module to a classical network to generate stochastic solutions of the SR samples. In general diffusion models have impressive performance to generate natural images but can have a lack of realism to generate high-frequency textures (as confirmed below by our comparative experiments). Other networks can provide results with some artefacts, even though this may be overcome using parametric priors [7].

Another approach is to apply reference-based SR [17, 16, 72, 6, 45]. In this setting the LR input is provided with a companion HR image that conveys information on the content of the unknown HR solution, e.g. by providing a HR image of the content from another viewpoint. Particularly relevant for this paper, in the context of super-resolution of stationary texture images, the patch distribution of the reference image can be exploited by minimizing the the Wasserstein distance between HR output patches and the reference patches [28]. This variational approach, coined Wasserstein Patch Prior (WPP), exploits important statistics proved to be relevant for texture synthesis [27, 21, 31, 32]. Let us mention that WPP can be used as a training loss for conditional generative models [2] and made more robust using semi-unbalanced optimal transport [48, 49]. The WPP framework provides adaptative statistical guarantees to the proven efficiency of patch-based approaches [73, 68], which is critical for the SR of texture images. Other SR patch-based methods produce an output HR image from the LR image input by learning local adversarial regularizers [54] or using patch normalizing flow regularizers [3].

In this work, we will solely focus on a precise class of textures, the so-called Gaussian microtextures [22]. This is a simple texture model that assumes that the texture is stationary (statistically invariant by translation) and follows a Gaussian distribution. Even though the class of texture images well-reproduced by this model is restricted, the Gaussian modeling allows for many probabilistic and statistical tools to be used for processing these textures. The textures can be summarized by a local texton [12, 19], can be generated on arbitrary large continuous domains [18, 20], extended to video and visually mixed using Wasserstein barycenters between Gaussian distributions [67]. Closely related to our work, the Gaussian modeling allows for solving texture inpainting via Gaussian conditional sampling based on kriging formulas [25, 24]. This last work provides an iterative algorithm based on conjugate gradient descent (CGD) that we extend to SR and use as a reference for evaluating our faster "Gaussian SR" solution. A preliminary version of the present work has been presented at ICASSP 2023 [53].

1.2 Contributions

Our approach builds on the kriging framework used for Gaussian inpainting [25, 24]. We further exploit the stationarity of the zoom-out operator to obtain an exact and fast convolution-based algorithm for stochastic SR of grayscale Gaussian textures. Due to correlation between color chanels, this algorithm does not translate straightforwardly to RGB textures, but we propose an approximate algorithm for RGB textures that proves to be experimentally close to the exact but time-consuming algorithm. Finally, on a practical viewpoint, we extend our approach to the context of SR given a reference HR image, and show that, when applied to Gaussian textures, our fast Gaussian SR algorithm is both perceptually better and order of magnitudes faster than recent deep learning or optimal transport-based methods. While our approach has several inherent limitations that we discuss and illustrate, our strong experimental results show that it is of practical interest when dealing with simple microtextures.

1.3 Plan of the paper

The plan of the paper is as follows. In Section 2 we present our framework and remind results about Gaussian conditional simulation. Then, we detail an exact, stable and efficient simulation procedure of conditional SR for grayscale Gaussian microtextures. In Section 3, we propose an approximate algorithm for stochastic SR of RGB microtextures and compare it with the CGD-based reference approach. The extension of our sampling algorithm in the practical settings of SR given a reference image is discussed in Section 4 where we compare our results with state-of-the-art methods. We also show that our algorithm extends to more general linear operators and discuss the robustness of our algorithm in the presence of measurement noise. This comparaison leads to a discussion on adapted metrics to evaluate textures SR. Finally, we document the shortcomings of our approach in Section 5 and conclude our paper in Section 6.

1.4 Notation

Let $M, N > 2$ be the size of the HR images and $r > 1$ a zoom-out factor such that the integers $M/r, N/r$ are the size of the LR images (it is assumed that r divides both M and N). We write $[k] = \{0, \dots, k-1\}$ for any integer $k \geqslant 1$, $\Omega_{M,N} = [M] \times [N]$ denotes the pixel grid, and all the images are extended on $\mathbb{Z}^2$ by periodization. For $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{\Omega_{M,N}}$, $\boldsymbol{u} \star \boldsymbol{v}$ designates the discrete and periodic convolution defined by $(\boldsymbol{u} \star \boldsymbol{v})(x) = \sum_{y \in \Omega_{M,N}} \boldsymbol{u}\big(x - y\big)\boldsymbol{v}(y)$ for $x \in \Omega_{M,N}$. Given a kernel $\boldsymbol{t} \in \mathbb{R}^{\Omega_{M,N}}$, let us denote $\boldsymbol{C_t} \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M,N}}$ the matrix associated with the convolution by $\boldsymbol{t}$ such that for $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$, $\boldsymbol{C_t u} = \boldsymbol{t} \star \boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$. $\boldsymbol{I}_{\Omega_{M,N}}$ denotes the identity matrix on $\Omega_{M,N}$ while $\mathbf{1}_{\Omega_{M,N}}$ denotes the constant image of $\mathbb{R}^{\Omega_{M,N}}$ with value 1. $\check{\boldsymbol{u}}$ stands for the symmetric image of $\boldsymbol{u}$ defined by for $\check{\boldsymbol{u}}(x) = \boldsymbol{u}(-x)$ for $x \in \Omega_{M,N}$. We express by $\mathscr{F}(\boldsymbol{u})$ or $\widehat{\boldsymbol{u}}$ the Discrete Fourier Transform (DFT) of $\boldsymbol{u}$ defined by $\widehat{\boldsymbol{u}}(x) = \sum_{y \in \Omega_{M,N}} \boldsymbol{u}(y) \exp(-\frac{2i\pi x_1 y_1}{M}) \exp(-\frac{2i\pi x_2 y_2}{N})$ for $x \in \Omega_{M,N}$. Let us recall that $\mathscr{F}$ is invertible with $\mathscr{F}^{-1} = \frac{1}{MN}\overline{\mathscr{F}}$, $\mathscr{F}(\check{\boldsymbol{u}}) = \overline{\mathscr{F}(\boldsymbol{u})}$ and $\mathscr{F}(\boldsymbol{u} \star \boldsymbol{v}) = \mathscr{F}(\boldsymbol{u}) \odot \mathscr{F}(\boldsymbol{v})$ where $\odot$ denotes the componentwise product. We denote by $\mathscr{F}_r$ the Fourier transform of images of $\mathbb{R}^{\Omega_{M/r,N/r}}$.

2 The conditional super-resolution of grayscale Gaussian microtextures

In this section we solve the problem of Stochastic super-resolution of grayscale Gaussian microtextures using a known zoom-out operator given by a convolution. Exploiting both the stationarity of the LR measurements and the HR texture model, our main result states that one can exactly sample new HR microtextures by applying a specific convolution operator. We first recall the specifics of the zoom-out operator and the ADSN microtexture model. Then we recall how solving for kriging coefficients allows for conditional Gaussian simulation and establish our main result.

2.1 The zoom-out operator

The most common zoom-out operator used in the SR literature is the Matlab bicubic zoom-out operator provided by the function `imresize` [43, 65, 4]. Let us denote $\boldsymbol{A}$ this bicubic zoom-out operator by a factor r. It can be described as a convolution followed by a subsampling $\boldsymbol{S}$ with stride r such that for $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$, $\boldsymbol{Su} \in \mathbb{R}^{\Omega_{M/r,N/r}}$ and for $x \in \Omega_{M/r,N/r}$, $(\boldsymbol{Su})(x) = \boldsymbol{u}(rx)$ and for $x \in \Omega_{M/r,N/r}, \boldsymbol{v} \in \mathbb{R}^{\Omega_{M/r,N/r}}$, $(\boldsymbol{S}^T\boldsymbol{v})(rx) = \boldsymbol{v}(x)$, see e.g. [4]. We denote by $\boldsymbol{c}$ the kernel associated with the bicubic kernel $\boldsymbol{A} = \boldsymbol{SC_c}$. In the remaining of the paper, we study the following inverse problem

$$\boldsymbol{u}_{\mathrm{LR}} = \boldsymbol{A}\boldsymbol{u}_{\mathrm{HR}} \tag{1}$$

where $\boldsymbol{u}_{\mathrm{LR}}$ is the observed low resolution version of an unknown HR image $\boldsymbol{u}_{\mathrm{HR}}$. This problem is ill-posed, its space of solutions is a high-dimensional affine subspace, and most such solutions are not desirable such that blurry HR versions of the LR image $\boldsymbol{u}_{\mathrm{LR}}$. Our main result provides an efficient sampler of solutions of Equation (1) under the assumption that $\boldsymbol{u}_{\mathrm{HR}}$ follows an ADSN distribution, that is, a Gaussian microtexture model we recall below.

2.2 The ADSN model associated with a grayscale image

Given a grayscale image $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ with mean grayscale $m \in \mathbb{R}$, one defines the Asymptotic Discrete Spot Noise (ADSN) associated with $\boldsymbol{u}$, $\mathrm{ADSN}(\boldsymbol{u})$ as the distribution of $m\mathbf{1}_{\Omega_{M,N}} + \boldsymbol{t} \star \boldsymbol{w}$ where $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m\mathbf{1}_{\Omega_{M,N}})$ is called the texton associated with $\boldsymbol{u}$ and $\boldsymbol{w} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{I}_{\Omega_{M,N}})$ [22]. Let us recall that in practice it is preferable to replace $\boldsymbol{u}$ by its periodic component [50] to avoid boundary issues [22]. $\mathrm{ADSN}(\boldsymbol{u})$ is a Gaussian distribution with mean $m\mathbf{1}_{\Omega_{M,N}}$ and covariance matrix $\boldsymbol{\Gamma} = \boldsymbol{C_t C_t}^T = \boldsymbol{C}_{\boldsymbol{t} \star \check{\boldsymbol{t}}} \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M,N}}$. Samples can be generated fastly executing the convolution with the Fast Fourier Transform (FFT). $\mathrm{ADSN}(\boldsymbol{u})$ is stationary, that is, for $\boldsymbol{X} \sim \mathrm{ADSN}(\boldsymbol{u})$ and any $y \in \mathbb{Z}^2$, $\boldsymbol{X}(. - y)$ has the same distribution as $\boldsymbol{X}$. This implies that the pixelwise variance of this distribution is constant. This restricts the application of ADSN model to microtextures that do not have structures. Practically, substracting and adding the common mean gray-level to LR and HR images is considered as a preprocessing and a post-processing. In what follows, we will often consider the centered distribution $\boldsymbol{t} \star \boldsymbol{w} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ that will be called $\mathrm{ADSN}(\boldsymbol{u})$ with a slight abuse of notation.

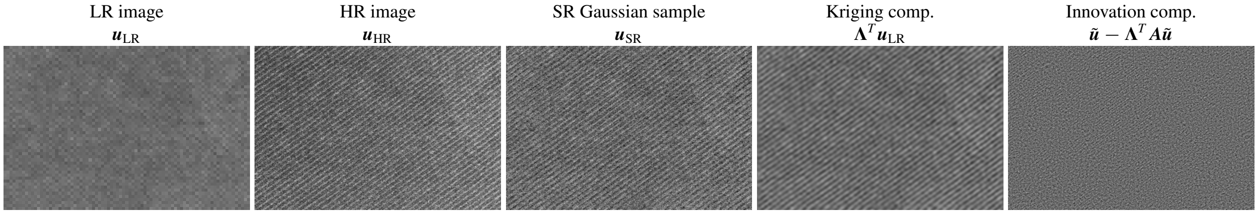


Figure 1: Super resolution of a Gaussian texture using kriging. From left to right: LR image, original HR image, conditional Gaussian sample and its corresponding kriging and innovation components. HR images size is 416×640 and the zoom-out factor is $r = 8$. The sample is the sum of the kriging component and the innovation component. Grayscale mean is added to all components for visualization. The deterministic kriging component is a kind of pseudo-inversion of the zoom-out operator applied to $\boldsymbol{u}_{\mathrm{LR}}$ that benefits from the knowledge of the covariance of the texture model. The innovation component provides an independent texture grain that is sampled for each realization of $\boldsymbol{u}_{\mathrm{SR}}$.

2.3 The kriging reasoning

Let $\boldsymbol{X} \in \mathbb{R}^{\Omega_{M,N}}$ be a random vector following a Gaussian multivariate law $\mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ with $\boldsymbol{\Gamma} \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M,N}}$ and let $\boldsymbol{A} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ be a linear operator. The simulation of $\boldsymbol{X}$ knowing $\boldsymbol{AX}$ can be computed using the following theorem, recalled in [24], which expresses the link between orthogonality and independence for Gaussian vectors.

Theorem 1 (Contidional Gaussian simulation)**.** *Let $\boldsymbol{X} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\boldsymbol{A}$ be a linear operator. The two Gaussian vectors $\mathbb{E}(\boldsymbol{X}|\boldsymbol{AX})$ and $\boldsymbol{X} - \mathbb{E}(\boldsymbol{X}|\boldsymbol{AX})$ are independent. Consequently, if $\tilde{\boldsymbol{X}}$ is independent of $\boldsymbol{X}$ with the same distribution then $\mathbb{E}(\boldsymbol{X}|\boldsymbol{AX}) + [\tilde{\boldsymbol{X}} - \mathbb{E}(\tilde{\boldsymbol{X}}|\boldsymbol{A}\tilde{\boldsymbol{X}})]$ has the same distribution as $\boldsymbol{X}$ knowing $\boldsymbol{AX}$.*

Furthermore, in the Gaussian context, if the distribution is zero-mean, the relation between the conditional expectation $\mathbb{E}(\boldsymbol{X}|\boldsymbol{AX})$ and $\boldsymbol{AX}$ is linear, as detailed in the following.

Theorem 2 (Gaussian kriging)**.** *Let $\boldsymbol{X} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ be a* ***Gaussian*** *vector with* ***zero mean*** *and $\boldsymbol{A} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ be a linear operator. There exists $\boldsymbol{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ such that $\mathbb{E}(\boldsymbol{X}|\boldsymbol{AX}) = \boldsymbol{\Lambda}^T\boldsymbol{AX}$ if and only if $\boldsymbol{\Lambda}$ verifies the matrix equation*

$$\boldsymbol{A\Gamma A}^T\boldsymbol{\Lambda} = \boldsymbol{A\Gamma}. \tag{2}$$

Theorem 2 is a standard [41]. Yet since solving Equation (2) is central in what follows, we give a detailed proof in Appendix A. A matrix $\boldsymbol{\Lambda}$ is called a kriging matrix and Equation (2) is called the kriging equation. Note that if $\boldsymbol{A}$ is invertible, a trivial solution for $\boldsymbol{\Lambda}^T$ is $\boldsymbol{A}^{-1}$. In our context of super-resolution, $\boldsymbol{A}$ is not invertible and we are given an LR image $\boldsymbol{u}_{\mathrm{LR}} = \boldsymbol{A}\boldsymbol{u}_{\mathrm{HR}}$ with $\boldsymbol{u}_{\mathrm{HR}}$ unknown. Supposing that $\boldsymbol{u}_{\mathrm{HR}}$ is a realization of a law $\mathrm{ADSN}(\boldsymbol{u}) = \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$, we would like to sample HR images $\boldsymbol{u}_{\mathrm{SR}}$ such that $\boldsymbol{u}_{\mathrm{SR}}$ follows $\mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ conditioned on $\boldsymbol{A}\boldsymbol{u}_{\mathrm{SR}} = \boldsymbol{u}_{\mathrm{LR}}$. Using Theorem 1 and Theorem 2, we aim at computing a HR sample image

$$\boldsymbol{u}_{\mathrm{SR}} = \boldsymbol{\Lambda}^T\boldsymbol{u}_{\mathrm{LR}} + \tilde{\boldsymbol{u}} - \boldsymbol{\Lambda}^T\boldsymbol{A}\tilde{\boldsymbol{u}} \tag{3}$$

where $\tilde{\boldsymbol{u}} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\boldsymbol{\Lambda}$ verifies the kriging Equation (2). Note that $\boldsymbol{u}_{\mathrm{SR}}$ is the sum of a deterministic term $\boldsymbol{\Lambda}^T\boldsymbol{u}_{\mathrm{LR}}$, called the *kriging component*, and a stochastic term $\tilde{\boldsymbol{u}} - \boldsymbol{\Lambda}^T\tilde{\boldsymbol{u}}$ not depending on $\boldsymbol{u}_{\mathrm{LR}}$, called the *innovation component*. As presented in Figure 1, the kriging component is a HR blurred version of the LR image with added covariance information while the innovation component provides independently the granular aspect of the microtexture. In general, the kriging equation needs to be solved by an iterative method, even with the stationarity assumption of the Gaussian law [24]. Due to the stationarity of the Gaussian law $\mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and the convolution form of $\boldsymbol{A}$, we show below that it is possible to solve Equation (2) exactly and fastly in the Fourier domain.

2.4 Solving the kriging equation in the super-resolution context

We show in this section that in our SR context, the kriging equation can be solved efficiently with a non-iterative algorithm.

Equation (2) may have several solutions and we only consider the specific solution $\boldsymbol{\Lambda} = (\boldsymbol{A\Gamma A}^T)^{\dagger}\boldsymbol{A\Gamma}$ where $(\boldsymbol{A\Gamma A}^T)^{\dagger}$ is the pseudo-inverse of $\boldsymbol{A\Gamma A}^T$. The main practical issue is now to compute efficiently $\boldsymbol{\Lambda}^T\boldsymbol{v}$ for a given LR image $\boldsymbol{v} \in \mathbb{R}^{\Omega_{M/r,N/r}}$. As shown by the following proposition, this simply corresponds to applying a specific convolution.

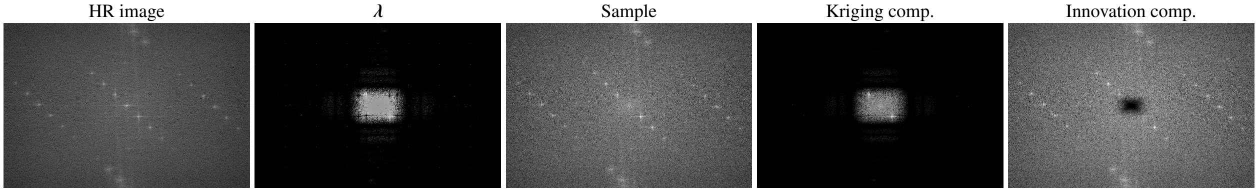


Figure 2: Observation of the modulus logscale of DFT of the different components from Figure 1. $\boldsymbol{\lambda}$ is the kernel associated with the kriging matrix $\boldsymbol{\Lambda}^T$ of Proposition 1. Note that the kriging component retrieves low frequencies from LR image that are completed with the high frequencies of the innovation component.

Proposition 1 (Kriging as a convolution)**.** *Let $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{C}_{\boldsymbol{c}}$ and let $\boldsymbol{\Gamma} = \boldsymbol{C}_{\boldsymbol{t}}\boldsymbol{C}_{\boldsymbol{t}}^T = \boldsymbol{C}_{\boldsymbol{t}\star\check{\boldsymbol{t}}}$ be the covariance of a Gaussian distribution* ADSN$(\boldsymbol{u})$ *with associated texton $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m\mathbf{1}_{\Omega_{M,N}})$. $\boldsymbol{\Lambda} = \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger \boldsymbol{A}\boldsymbol{\Gamma}$ is an exact solution of Equation* (2) *and for all $\boldsymbol{v} \in \mathbb{R}^{\Omega_{M/r,N/r}}$,*

$$\boldsymbol{\Lambda}^T\boldsymbol{v} = \boldsymbol{\lambda} \star (\boldsymbol{S}^T\boldsymbol{v}) \tag{4}$$

where $\boldsymbol{\lambda} = \boldsymbol{t} \star \check{\boldsymbol{t}} \star \check{\boldsymbol{c}} \star (\boldsymbol{S}^T\boldsymbol{\kappa}^\dagger)$ with $\boldsymbol{\kappa} = \boldsymbol{S}(\boldsymbol{t} \star \check{\boldsymbol{t}} \star \boldsymbol{c} \star \check{\boldsymbol{c}}) \in \mathbb{R}^{\Omega_{M/r,N/r}}$ and $\boldsymbol{\kappa}^\dagger$ the convolution kernel defined in Fourier domain by

$$\widehat{\boldsymbol{\kappa}}^\dagger(\omega) = \begin{cases} \frac{1}{\widehat{\boldsymbol{\kappa}}(\omega)} & \textit{if } \widehat{\boldsymbol{\kappa}}(\omega) \neq 0, \\ 0 & \textit{otherwise}, \end{cases} \qquad \omega \in \mathbb{R}^{\Omega_{M/r,N/r}}.$$

Proof. Equation (2) is a normal equation associated with the least squares problem $\operatorname{argmin}_{\boldsymbol{\Lambda}\in\mathbb{R}^{\Omega_{M/r,N/r}\times\Omega_{M,N}}} \left\|\boldsymbol{C}_{\boldsymbol{t}}^T\boldsymbol{A}^T\boldsymbol{\Lambda} - \boldsymbol{C}_{\boldsymbol{t}}^T\right\|_2^2$. Consequently, $\boldsymbol{\Lambda} = \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger \boldsymbol{A}\boldsymbol{\Gamma}$ is one of the solutions of Equation (2). Then, we use the following elementary lemma (proved in Appendix B for completeness).

Lemma 1 (Convolution and subsampling)**.**

1. *If $\boldsymbol{C}_{\boldsymbol{\alpha}}$ is the convolution on $\Omega_{M,N}$ by the kernel $\boldsymbol{\alpha} \in \Omega_{M,N}$, $\boldsymbol{S}\boldsymbol{C}_{\boldsymbol{\alpha}}\boldsymbol{S}^T$ is the convolution on $\Omega_{M/r,N/r}$ by the kernel $\boldsymbol{S\alpha}$, that is, $\boldsymbol{S}\boldsymbol{C}_{\boldsymbol{\alpha}}\boldsymbol{S}^T = \boldsymbol{C}_{\boldsymbol{S\alpha}}$.*
2. *If $\boldsymbol{C}_{\boldsymbol{\beta}}$ is a convolution on $\Omega_{M/r,N/r}$ by the kernel $\boldsymbol{\beta} \in \Omega_{M/r,N/r}$, $\boldsymbol{S}^T\boldsymbol{C}_{\boldsymbol{\beta}} = \boldsymbol{C}_{\boldsymbol{S}^T\boldsymbol{\beta}}\boldsymbol{S}^T$.*

By Lemma 1, $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T = \boldsymbol{S}(\boldsymbol{C}_{\boldsymbol{c}}\boldsymbol{\Gamma}\boldsymbol{C}_{\boldsymbol{c}}^T)\boldsymbol{S}^T$ is a convolution matrix associated with the kernel $\boldsymbol{\kappa} = \boldsymbol{S}(\boldsymbol{t}\star\check{\boldsymbol{t}}\star\boldsymbol{c}\star\check{\boldsymbol{c}}) \in \mathbb{R}^{\Omega_{M/r,N/r}}$. Moreover, the pseudo-inverse of a convolution is also a convolution. $\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger$ is a convolution associated with the kernel $\boldsymbol{\kappa}^\dagger$ such that for $\omega \in \mathbb{R}^{\Omega_{M/r,N/r}}, \widehat{\boldsymbol{\kappa}}^\dagger(\omega) = \begin{cases} \frac{1}{\widehat{\boldsymbol{\kappa}}(\omega)} & \text{if } \widehat{\boldsymbol{\kappa}}(\omega) \neq 0 \\ 0 & \text{otherwise.} \end{cases}$. Also by Lemma 1, $\boldsymbol{S}^T(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T)^\dagger = \boldsymbol{S}^T\boldsymbol{C}_{\boldsymbol{\kappa}^\dagger} = \boldsymbol{C}_{\boldsymbol{S}^T\boldsymbol{\kappa}^\dagger}\boldsymbol{S}^T$. Consequently,

$$\boldsymbol{\Lambda}^T = \boldsymbol{\Gamma}^T\boldsymbol{A}^T\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger = \boldsymbol{\Gamma}^T\boldsymbol{C}_{\boldsymbol{c}}^T\boldsymbol{S}^T\boldsymbol{C}_{\boldsymbol{\kappa}^\dagger} = \boldsymbol{\Gamma}^T\boldsymbol{C}_{\boldsymbol{c}}^T\boldsymbol{C}_{\boldsymbol{S}^T\boldsymbol{\kappa}^\dagger}\boldsymbol{S}^T = \boldsymbol{C}_{\boldsymbol{\lambda}}\boldsymbol{S}^T$$

where $\boldsymbol{\lambda} = \boldsymbol{t} \star \check{\boldsymbol{t}} \star \check{\boldsymbol{c}} \star \boldsymbol{S}^T\boldsymbol{\kappa}^\dagger$. □

In what follows we refer to $\boldsymbol{\lambda}$ as the *kriging kernel*. Figure 2 shows the DFT of the kernel $\boldsymbol{\lambda}$ associated with the images of Figure 1. In contrast with a non-adaptative pseudo-inversion of the bicubic kernel that would be isotropic, we can observe that the kernel $\boldsymbol{\lambda}$ is adapted to the covariance structure of the texture and that it amplifies its characteristic frequencies.

To compute SR samples $\boldsymbol{u}_{\mathrm{SR}}$ given by Equation (3), we need to compute $\boldsymbol{\lambda} \in \mathbb{R}^{\Omega_{M,N}}$ the kernel associated with $\boldsymbol{\Lambda}^T$ as described in Proposition 1 and then apply $\boldsymbol{\Lambda}^T$ as described in Equation (3). These two steps can be done fastly in the Fourier domain. The exact and fast corresponding procedure is described in Algorithm 1. To generate several samples, one only needs to rerun the second part of the algorithm.

The application of $\boldsymbol{\Lambda}^T$ implies the division by $\widehat{\boldsymbol{\kappa}}$ in the Fourier domain. However, in practice, this division can be unstable since a non-zero low value of $\widehat{\boldsymbol{\kappa}}$ amplifies its corresponding frequency. Actually, in our context, $\boldsymbol{\Lambda}^T$ is applied to LR versions of realizations of an ADSN model that are $\boldsymbol{v} = \boldsymbol{A}(\boldsymbol{t} \star \boldsymbol{w})$ with $\boldsymbol{w} \in \mathbb{R}^{\Omega_{M,N}}$ an unknown white Gaussian noise. Proposition 2 ensures that the application of $\boldsymbol{\Lambda}^T$ is stable when applied to images that comply with this assumption. This stability case is no more valid in the presence of measurement noise discussed in Section 4.3.

Algorithm 1 Super-resolution sampling for grayscale images with known Gaussian model

```
Input: An image $\boldsymbol{u}_{\mathrm{LR}} \in \mathbb{R}^{\Omega_{M/r,N/r}}$, $r$ the zoom factor, $\boldsymbol{t}$ the convolution kernel of the ADSN model, $\boldsymbol{c}$ the kernel of the convolution of the zoom-out operator $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{C}_{\boldsymbol{c}}$.
Preprocessing:
Compute the grayscale mean $m$ from $\boldsymbol{u}_{\mathrm{LR}}$ and set $\boldsymbol{u}_{\mathrm{LR}} := \boldsymbol{u}_{\mathrm{LR}} - m\mathbf{1}_{\Omega_{M,N}}$
Step 1: Computation of the kriging kernel
Store the DFT transform of the kernel $\boldsymbol{\lambda} = \boldsymbol{t} \star \check{\boldsymbol{t}} \star \check{\boldsymbol{c}} \star \boldsymbol{S}^T(\boldsymbol{\kappa}^\dagger)$
Step 2: Simulation of $\boldsymbol{u}_{\mathrm{SR}}$
Sample $\tilde{\boldsymbol{u}} = \boldsymbol{t} \star \boldsymbol{w}$ where $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{\Omega_{M,N}})$
Compute $\boldsymbol{u}_{\mathrm{SR}} = \boldsymbol{\lambda} \star \boldsymbol{S}^T(\boldsymbol{u}_{\mathrm{LR}} - \boldsymbol{A}\tilde{\boldsymbol{u}}) + \tilde{\boldsymbol{u}}$
Postprocessing:
Output: $m\mathbf{1}_{\Omega_{M,N}} + \boldsymbol{u}_{\mathrm{SR}}$
```

Proposition 2 (Stability of the kriging operator on the subspace of the LR ADSN samples)**.** *Let* $\boldsymbol{\Lambda}^T = \boldsymbol{\Gamma}\boldsymbol{A}^T\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M/r,N/r}}$*. Then,*

$$\forall \boldsymbol{w} \in \mathbb{R}^{\Omega_{M,N}}, \left\|\boldsymbol{\Lambda}^T\boldsymbol{A}(\boldsymbol{t} \star \boldsymbol{w})\right\|_2 \leqslant \|\boldsymbol{C}_{\boldsymbol{t}}\|_2\|\boldsymbol{w}\|_2 \leqslant \|\boldsymbol{t}\|_1\|\boldsymbol{w}\|_2. \tag{5}$$

Proof. As a reminder [52], let $\boldsymbol{M}$ being a real-valued matrix,

$$\boldsymbol{M}^\dagger = \left(\boldsymbol{M}^T\boldsymbol{M}\right)^\dagger \boldsymbol{M}^T$$

With $\boldsymbol{M} = (\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^T$,

$$(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^{T\dagger} = \left(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}}(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^T\right)^\dagger \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}} = \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}}$$

Consequently, for $\boldsymbol{w} \in \mathbb{R}^{\Omega_{M,N}}$,

$$\boldsymbol{\Lambda}^T\boldsymbol{A}(\boldsymbol{t} \star \boldsymbol{w}) = \boldsymbol{\Gamma}\boldsymbol{A}^T\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}}\boldsymbol{w} = \boldsymbol{C}_{\boldsymbol{t}}(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^T\left((\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^T\right)^\dagger \boldsymbol{w}$$

and

$$\left\|\boldsymbol{\Lambda}^T\boldsymbol{A}(\boldsymbol{t} \star \boldsymbol{w})\right\|_2 \leqslant \|\boldsymbol{C}_{\boldsymbol{t}}\|_2 \underbrace{\left\|(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^T\left((\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{t}})^T\right)^\dagger\right\|_2}_{\leqslant 1} \|\boldsymbol{w}\|_2 \leqslant \|\boldsymbol{C}_{\boldsymbol{t}}\|_2\|\boldsymbol{w}\|_2 \leqslant \|\boldsymbol{t}\|_1\|\boldsymbol{w}\|_2,$$

using that for any matrix $\boldsymbol{M}$ one has $\|\boldsymbol{M}\boldsymbol{M}^\dagger\|_2 \leqslant 1$. □

3 The conditional super-resolution of RGB Gaussian microtextures

3.1 Description of the framework and notation

We denote by $3\Omega_{M,N}$ the space of the RGB images and by $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$ the three chanels of a given RGB image $\boldsymbol{u} \in \mathbb{R}^{3\Omega_{M,N}}$. $\widehat{\boldsymbol{u}}$ designates the Discrete Fourier transform of $\boldsymbol{u}$ defined as the 2D DFT of each chanel. The associated texton of an RGB image $\boldsymbol{u}$ is $\boldsymbol{t} = \frac{1}{\sqrt{NM}}(\boldsymbol{u} - \boldsymbol{m}) \in \mathbb{R}^{3\Omega_{M,N}}$ where $\boldsymbol{m} \in \mathbb{R}^{3\Omega_{M,N}}$ is the mean RGB color. The ADSN model ADSN$(\boldsymbol{u})$ is defined as the distribution of $\boldsymbol{m} + \begin{pmatrix} \boldsymbol{t}_1 \star \boldsymbol{w} \\ \boldsymbol{t}_2 \star \boldsymbol{w} \\ \boldsymbol{t}_3 \star \boldsymbol{w} \end{pmatrix}$ where $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{\Omega_{M,N}})$. Note that the white Gaussian noise is the same in each chanel [22]. ADSN$(\boldsymbol{u})$ is a multivariate Gaussian law associated with the covariance matrix $\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{C}_{\boldsymbol{t}_1} \\ \boldsymbol{C}_{\boldsymbol{t}_2} \\ \boldsymbol{C}_{\boldsymbol{t}_3} \end{pmatrix} \begin{pmatrix} \boldsymbol{C}_{\boldsymbol{t}_1} \\ \boldsymbol{C}_{\boldsymbol{t}_2} \\ \boldsymbol{C}_{\boldsymbol{t}_3} \end{pmatrix}^T \in \mathbb{R}^{3\Omega_{M,N} \times 3\Omega_{M,N}}$ which is the matrix of a multi-channel convolution, that is, each output chanel is a linear combination of 2D convolutions of the input RGB chanels. More precisely, for all $\boldsymbol{u} \in \mathbb{R}^{3\Omega_{M,N}}$,

$$(\boldsymbol{\Gamma}\boldsymbol{u})_i = \boldsymbol{t}_i \star \check{\boldsymbol{t}}_1 \star \boldsymbol{u}_1 + \boldsymbol{t}_i \star \check{\boldsymbol{t}}_2 \star \boldsymbol{u}_2 + \boldsymbol{t}_i \star \check{\boldsymbol{t}}_3 \star \boldsymbol{u}_3, \quad 1 \leqslant i \leqslant 3. \tag{6}$$

Unfortunately, this matrix is not diagonal in the Fourier basis which makes a major difference with the grayscale framework. We still denote by $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{C}_{\boldsymbol{c}}$ the zoom-out operator that acts on each chanel for an RGB image.

3.2 Approximating the kriging operator in the RGB case

To simulate conditional sample of the RGB ADSN model, the kriging equation Equation (2) and Lemma 1 are still valid, except that now $\boldsymbol{A\Gamma A}^T$ and $\left(\boldsymbol{A\Gamma A}^T\right)^\dagger$ are multi-chanel convolutions. However, while the pseudo-inverse of a 2D convolution is diagonal in the Fourier basis, this is not the case for a multi-channel convolution.

Computing the pseudo-inverse $\left(\boldsymbol{A\Gamma A}^T\right)^\dagger$ turns out to be a critical task. Even for small images, a direct computation of the pseudo-inverse of $\boldsymbol{A\Gamma A}^T$ with standard routines leads to instabilities. A more principled approach is to remark that for each $\boldsymbol{v} \in \mathbb{R}^{3\Omega_{M/r,N/r}}$ and each $\omega \in \Omega_{M/r,N/r}$, there exists a 3×3 matrix $\widehat{K}(\omega)$ such that $\mathscr{F}_r(\boldsymbol{A\Gamma A}^T\boldsymbol{v})(\omega) = \widehat{K}(\omega)\widehat{\boldsymbol{v}}(\omega)$, reducing the problem to computing one 3×3 pseudo-inverse for each frequency. Still, each matrix is close to be singular which leads to a high instability in practice.

In order to keep an algorithm which is as fast and as stable as in the grayscale case (Algorithm 1), we propose to make the following approximation: let $\boldsymbol{X} \in \mathbb{R}^{3\Omega_{M,N}}$, following the Gaussian law $\mathscr{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$,

$$\mathbb{E}\left[\boldsymbol{X}_i \mid \boldsymbol{AX}_1, \boldsymbol{AX}_2, \boldsymbol{AX}_3\right] \approx \mathbb{E}\left[\boldsymbol{X}_i \mid \boldsymbol{AX}_i\right], \quad 1 \leqslant i \leqslant 3. \tag{7}$$

The rational behind this approximation is that to reconstruct the chanel $\boldsymbol{X}_i$ from the three chanels $\boldsymbol{AX} = (\boldsymbol{AX}_1\boldsymbol{AX}_2\boldsymbol{AX}_3)$, the more relevant information is in $\boldsymbol{AX}_i$. This is a reasonable assumption with respect to the behavior of natural images and the form of the bicubic convolution kernel. This approximation amounts to use

$$\boldsymbol{\Lambda}_{\text{approx}} = \begin{pmatrix} \boldsymbol{\Lambda}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Lambda}_3 \end{pmatrix} \in \mathbb{R}^{3\Omega_{M/r,N/r} \times 3\Omega_{M,N}} \tag{8}$$

as an approximate solution of Equation (2) in the RGB setting, where for each $1 \leqslant i \leqslant 3$, $\boldsymbol{\Lambda}_i \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ is a solution of

$$\boldsymbol{A\Gamma}_i\boldsymbol{A}^T\boldsymbol{\Lambda}_i = \boldsymbol{A\Gamma}_i, \quad 1 \leqslant i \leqslant 3, \tag{9}$$

$\boldsymbol{\Gamma}_i$ being the covariance matrix of the ith chanel. While $\boldsymbol{\Lambda}_{\text{approx}}$ is not a solution of the kriging equation, it is the kriging matrix associated with an ADSN law for which the RGB chanels are uncorrelated (which is not an interesting model to generate textures). However, we apply this kriging operator $\boldsymbol{\Lambda}_{\text{approx}}^T$ to textures having properly correlated channels according to the ADSN model $\mathscr{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$. More precisely, a RGB Gaussian SR sample $\boldsymbol{u}_{\text{SR}}$ is defined as

$$\boldsymbol{u}_{\text{SR},i} = \boldsymbol{\Lambda}_i^T(\boldsymbol{u}_{\text{LR},i} - \boldsymbol{A}\tilde{\boldsymbol{u}}_i) + \tilde{\boldsymbol{u}}_i, \quad 1 \leqslant i \leqslant 3, \tag{10}$$

where, for $1 \leqslant i \leqslant 3$, $\tilde{\boldsymbol{u}}_i = \boldsymbol{t}_i \star \boldsymbol{w}$ with $\boldsymbol{w} \sim \mathscr{N}(\boldsymbol{0}, \boldsymbol{I}_{\Omega_{M,N}})$ the common noise used for each channel and the multiplication by $\boldsymbol{\Lambda}_i^T$ is done following Algorithm 1.

In the next section, we assess that our proposed approximation is harmless by evaluating it with an exact reference iterative alternative.

3.3 Comparison with the reference CGD algorithm

In [24], the authors propose to solve the kriging equation for Gaussian microtextures to resolve the inpainting by conditional simulation. To solve Equation (2), they apply a CGD to compute $\left(\boldsymbol{A\Gamma A}^T\right)^\dagger \varphi$ for a given $\varphi \in \mathbb{R}^{\Omega_{M/r,N/r}}$. For completeness, the algorithm is recalled in Appendix C. This is an iterative algorithm with a stopping criteria ε and a given number of iterations. The output ψ of CGD tends to minimize

$$\left\|(\boldsymbol{A\Gamma A}^T)^2\varphi - (\boldsymbol{A\Gamma A}^T)\psi\right\|_2 \tag{11}$$

which is the residual for the normal equation associated with Equation (2). We consider CGD with a high number of 10^6 steps as a reference. The recommended number of steps for the inpainting problem is 10^2 for fast results and 10^3 for high-quality results [23] and our experiments show that these recommendations remain valid for super-resolution. Results from our fast convolution-based algorithm and this reference CGD-based algorithm are presented in Figure 3 for both the grayscale and RGB cases. The same realization of noise is used for the simulations so that both algorithm should produce the same image. Samples are visually similar for the two examples and when visualized as a sequence the texture details added by the CGD algorithm do not evolve after 10^3 steps.

Ìn Table 1, we compare the samples generated with our algorithm and the CGD routine for different number of steps using the PSNR with respect to the CGD output after 10^6 iterations and the CGD residual as metrics. For the

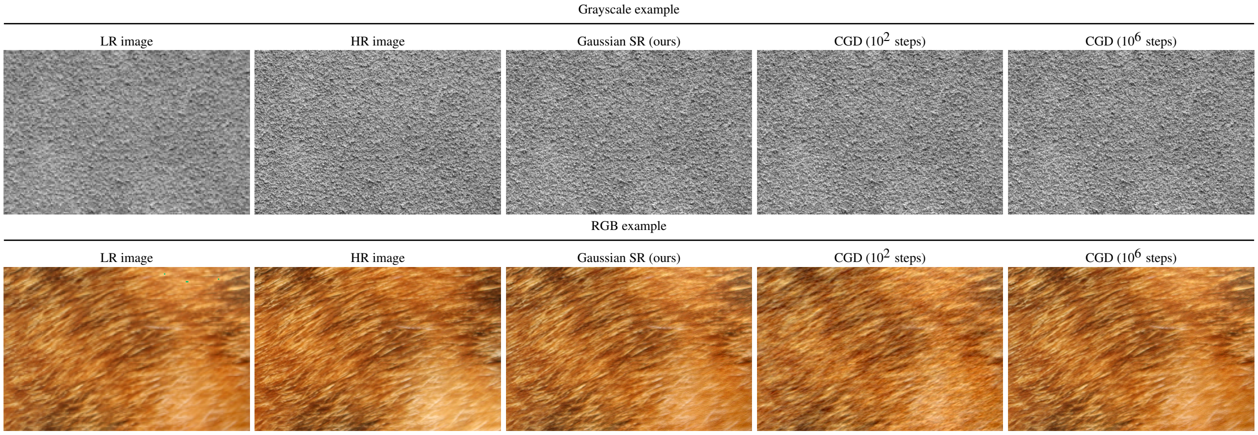


Figure 3: Comparison of our direct algorithm (Gaussian SR) and the iterative CGD-based algorithm for grayscale (first row) and RGB (second row) images. The HR image size is 512×768, the SR factor is $r = 4$, and the same noise realization is used to compute the innovation components to allow visual comparison. The two methods provide similar results even for the RGB case for which our algorithm is not exact.

Comparison with the reference CGD algorithm						
	Grayscale image			RGB image		
	Residual	Time(s) (CPU)	PSNR w.r.t CGD (10^6 st.)	Residual	Time(s) (CPU)	PSNR w.r.t CGD (10^6 st.)
Gaussian SR	1.32E−16	0.01	151.17	2.54E−1	0.01	37.94
CGD (10^2 steps)	2.74E−2	0.13	29.16	2.73E0	0.38	25.08
CGD (10^3 steps)	1.03E−3	0.76	47.49	3.78E−1	2.49	30.19
CGD (10^4 steps)	4.72E−5	7.44	67.60	1.36E−1	23.1	35.07
CGD (10^5 steps)	1.30E−8	81.3	145.31	2.37E−2	258	39.71
CGD (10^6 steps)	2.69E−42	749	-	4.97E−3	2588	-

Table 1: Quantitative evaluation of the results of Figure 3. The residual is the norm from Equation (11). The CGD is executed with different numbers of steps where 10^2 is the pre-set parameter in the online demo [23] for inpainting. The grayscale example confirms that our method is exact in this context. The RGB example shows that our approximation is harmless for RGB images. The different execution times show that producing samples with similar quality to our Gaussian SR samples using the CGD algorithm is four to five orders of magnitude slower.

grayscale image, we observe that our method is indeed exact with very low residual value. The high PSNR indicates that our direct Gaussian SR algorithm is more precise than CGD 10^5 steps.

In the color case, as expected, Gaussian SR is not exact and provides higher residual and PSNR values. Still the Gaussian SR samples are closer to CGD outputs with a number of steps between 10^4 and 10^5, which has been observed for several textures and for different zoom-out factors r. This validates our approach since the minor approximation we introduced is in practice negligible. Let us recall that the main objective of our approximation is to reduce the simulation time. As reported in Table 1, producing such an accurate RGB sample with the CGD algorithm is four to five orders of magnitude longer than with our direct convolution-based algorithm.

4 Gaussian SR in practice: Reference image and comparison with state-of-the-art

4.1 Gaussian SR with a reference image

To apply Gaussian SR algorithm, the ADSN distribution followed by $\boldsymbol{u}_{\mathrm{HR}}$ has to be known (via the computation of the associated texton $\boldsymbol{t}$). In the experiments of the previous section, the texton is computed from the ground truth HR image $\boldsymbol{u}_{\mathrm{HR}}$, making the algorithm impractical.

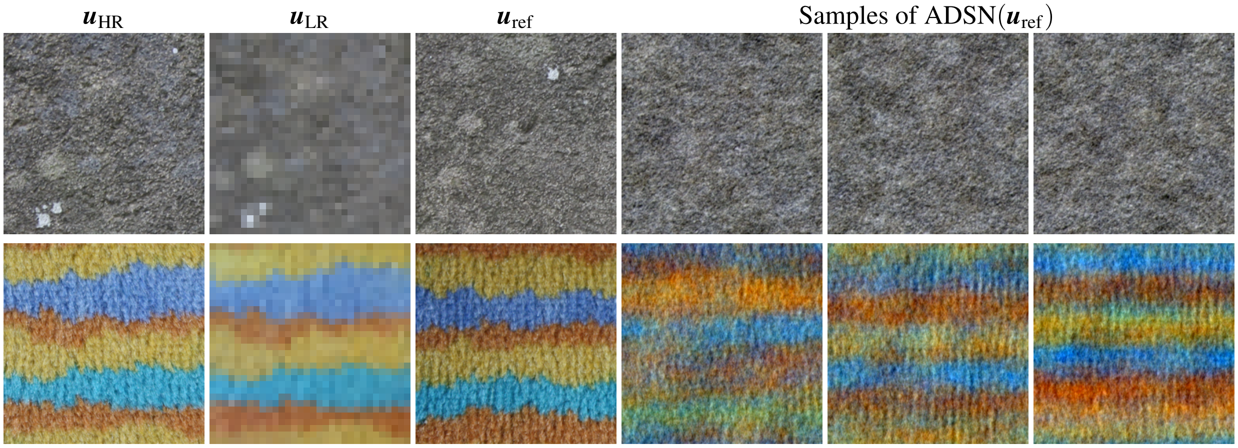


Figure 4: Example of the inputs of the Gaussian SR algorithm (LR image and Reference image) associated with a HR image (unknown) and samples of ADSN($\boldsymbol{u}_{\mathrm{ref}}$) used as a texture model. Note that the ADSN models are not visually faithful for image generation but they still provide a pertinent high frequency prior information to restore the textures for our stochastic SR routine.

From now on, following recent contributions on texture super-resolution [28, 2], we consider that the input image $\boldsymbol{u}_{\mathrm{LR}}$ is given with a companion reference HR image $\boldsymbol{u}_{\mathrm{ref}}$ such that the $\boldsymbol{u}_{\mathrm{HR}}$ is supposed to be a realization of the texture model ADSN($\boldsymbol{u}_{\mathrm{ref}}$). More precisely, in our context based on the ADSN model, Algorithm 1 will be called with input an LR image $\boldsymbol{u}_{\mathrm{LR}}$ to be upscaled and the texton $\boldsymbol{t}_{\mathrm{ref}} = \frac{1}{\sqrt{MN}}(\boldsymbol{u}_{\mathrm{ref}} - m\mathbf{1}_{\Omega_{M,N}})$ associated with the reference image $\boldsymbol{u}_{\mathrm{ref}}$, meaning that $\boldsymbol{u}_{\mathrm{HR}}$ is supposed to be a realization of the ADSN($\boldsymbol{u}_{\mathrm{ref}}$) law. This makes our routine a stochastic super-resolution algorithm with reference image. Two examples of input images (LR image and reference image) are given in Figure 4 along with samples of the distribution ADSN($\boldsymbol{u}_{\mathrm{ref}}$) used as a prior. Note that the relative bad quality of the ADSN samples is not an issue: this reference distribution only guides the restoration of high-frequencies.

Figure 5 presents illustrative results that show the practicality of this reference-based approach when the reference images are HR images of another part of the same material. Figure 5 also displays the kriging and innovation components and one can observe that the ADSN model from the reference image is good enough to compute an adapted kriging component. An implementation of the code is available online[1].

4.2 Comparison with other methods

Contrary to neural networks trained on a very large dataset of generic images, our method only works on the restricted area of Gaussian microtextures. This limited scope is a main drawback of Gaussian SR that cannot work on more generic images. Moreover, our method necessitates a HR reference image. Still it is interesting to compare our method with existing super-resolution algorithms, although one should keep in mind that these methods are designed to be used on a broader range of images.

We compare our method with other super-resolution methods on two microtextures in Figures 6 and 7 and table 2. Let us describe the concurrent algorithms and the metrics used for comparison.

Super-Resolution using Normalizing Flow (SRFlow[2]) [46], Denoising Diffusion Restoration Models (DDRM[3]) [37] and Manifold Constrained Gradient (MCG[4]) [9] are diverse super-resolution routines where several samples are proposed to solve the inverse problem 1. SRFlow is a normalizing flow network which has been trained during 5 days on a single NVIDIA V100 GPU with a general dataset of general images DIV2K [1] to solve the SR problem. It depends on a hyperparameter τ, the temperature, that modulates the variance of the latent space. MCG and DDRM work with a diffusion model that has been trained on the dataset ImageNet [11] for image generation during several days on GPU [13]. A step of attachment to degraded data is added in the generation to solve inverse problems such as SR. MCG proposes a gradient term and a projection on the image of $\boldsymbol{A}$ at each step to respect the constraint of

---

[1] `https://github.com/emilePi/Stochastic_SR_for_Gaussian_textures`

[2] Code and weights from `https://github.com/andreas128/SRFlow`

[3] Code and weights from `https://github.com/bahjat-kawar/ddrm`

[4] Code and weights from`https://github.com/HJ-harry/MCG_diffusion`

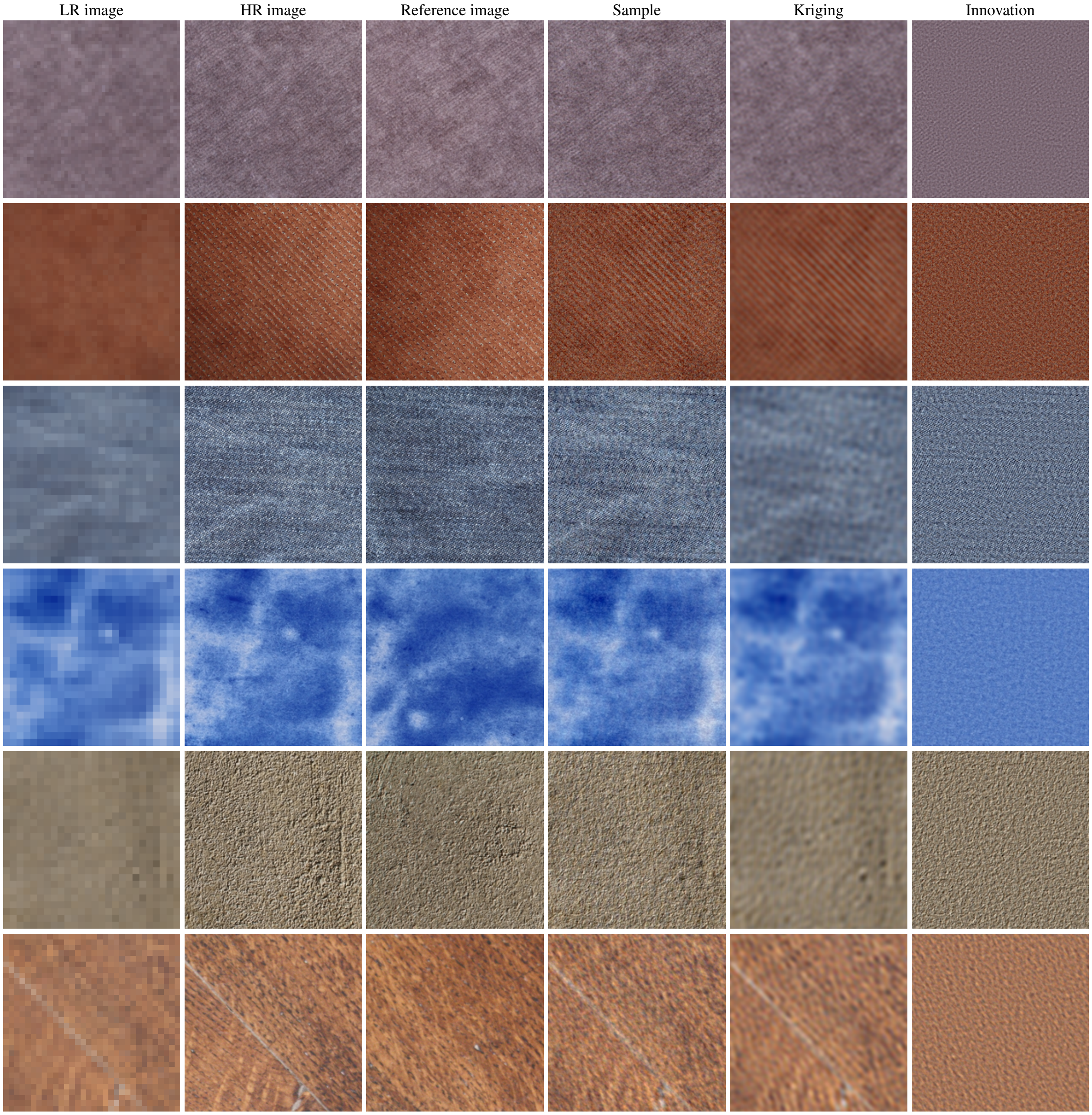


Figure 5: Illustration of Gaussian SR with a reference image (HR size is 208×208 and $r = 8$). Our method performs well on fabric textures and we can observe the grain provided by the innovation component for different types of texture. Some details are not recovered by our algorithm due to the stationarity assumption.

Equation (1). DDRM has an adaptative reverse process of the score-based model depending on the singular values of the operator $\boldsymbol{A}$. Similar to our method, Wasserstein Patch Prior (WPP[5]) [28] proposes to solve SR problem in the SISR setting using a reference image. It is a multiscale iterative algorithm that optimizes at several resolutions a optimal transport distance between patches of the proposed image and the ones of the reference image. In this comparative study, only WPP and Gaussian SR use the reference image while SRFlow, DDRM and MCG are trained on a larger dataset of generic images (and are not restricted to texture restoration).

To compare the methods, we use the four metrics Peak Signal to Noise Ratio (PSNR), LR-PSNR, Structural SIMi-

---

[5] `https://github.com/johertrich/Wasserstein_Patch_Prior`

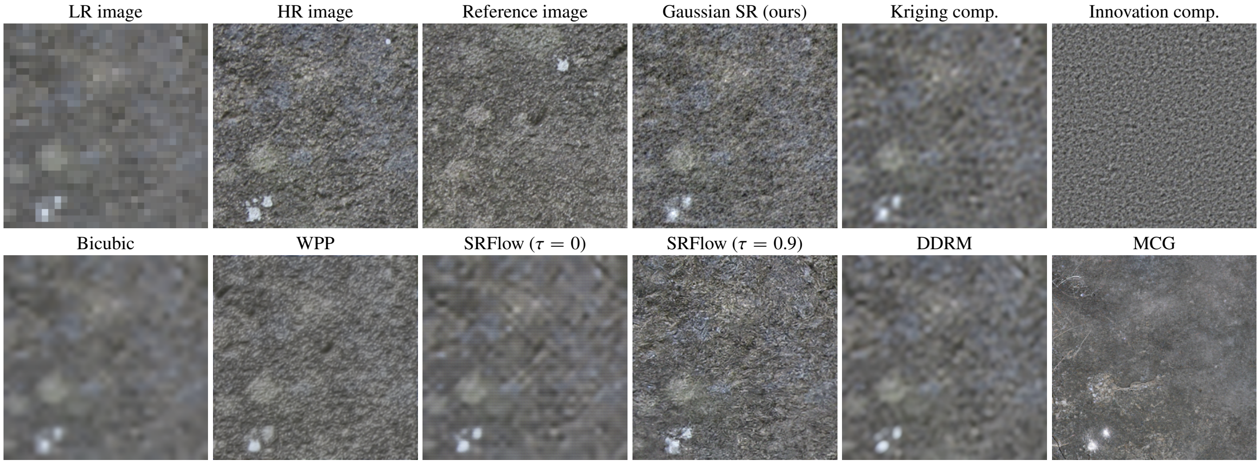


Figure 6: Comparison of our method with the stochastic SRflow, MCG and DDRM and with the SISR WPP that uses the reference image (The HR size is 256×256 and $r = 8$.). Note that our method is unable to provide the white spot in the wall that does not respect the stationarity assumption. However it faithfully conveys the granular aspect of the texture given by the reference image.

	Figure 6					Figure 7				
	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓	Time	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓	Time
Gaussian SR (ours)	17.05±0.04	159.24±0.04	0.08±0.01	**0.22±0.01**	**0.01**^{1}	19.14±0.09	**154.52±0.36**	0.20±0.01	**0.23±0.01**	**0.01**^{1}
Kriging comp.	18.76	**159.30**	0.11	0.75	-	21.42	154.47	**0.30**	0.52	-
Bicubic	20.74	37.61	0.18	0.76	-	20.74	38.35	0.28	0.66	-
WPP	19.99	40.44	0.14	0.27	44.0^2	19.30	24.16	0.22	0.26	64.0^2
SRFlow ($\tau = 0$)	21.44	54.61	**0.20**	0.70	0.22^2	21.64	51.63	0.29	0.54	0.23^2
SRFlow ($\tau = 0.9$)	18.29±0.36	55.13±0.15	0.12±0.01	0.30±0.03	0.19^2	18.21±0.53	54.02±0.23	0.16±0.02	0.39±0.06	0.20^2
DDRM	**21.58 ± 0.05**	56.78 ± 0.00	**0.20 ± 0.00**	0.70 ± 0.02	1.66^2	**22.44 ± 0.04**	55.90 ± 0.00	**0.30 ± 0.00**	0.55 ± 0.02	1.68^2
MCG	20.83 ± 0.52	30.84 ± 0.44	0.14 ± 0.01	0.61 ± 0.09	148^2	20.45 ± 0.27	30.08± 0.25	0.25 ± 0.01	0.58 ± 0.09	148^2

^{1}CPU,^{2}GPU Nvidia A100

Table 2: Quantitative comparison with the state-of-the-art methods results from Figures 6 and 7 realized on 100 samples for stochastic methods. Blurry results outperfoms on SSIM and PSNR metrics while our method provides the best perceptual LPIPS metric. Our method provides also a strong attachment to data, illustrated by the LR-PSNR metric. Our method is also faster, working on CPU.

larity(SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). The PSNR is the logarithmic scale of the mean-square error (MSE) between the HR image $\boldsymbol{u}_{\mathrm{HR}}$ and the solution proposed by the method. LR-PSNR is the PSNR between the LR image and the LR version of the solution and quantifies fidelity to the input data $\boldsymbol{u}_{\mathrm{LR}}$. The SSIM [66] quantifies the similarity between images relying on a luminance, a contrast and a structure terms. LPIPS [71] is a perceptual metric that compute the weighted squared distance between features of the two images of a classification network. The weights have been trained with respect to perceptual criteria.

Quantitatively, as observed in Figures 6 and 7, DDRM tends to produce blurry samples while MCG generates blurry images or very characteristic ones (e.g looking like satellite images). SRFlow with a positive temperature gives good textured solutions but with artefacts generally observed in networks' outputs. WPP generates textured but too smooth image. The variability of the stochastic SR algorithm is illustrated in Figure 15 in Appendix G. We can also observe that the different luminance of the reference image in Figure 7 influences the WPP result. Gaussian SR gives the desired perceptual grain of the textures. However, it does not retrieve correctly the details, as it can be observed with the white spot in Figure 6.

Table 2 shows that our method is faster and have the best LPIPS on each image, followed by WPP. Note that Gaussian SR works on CPU. Our method proposes also a strong data attachment, illustrated by the high LR-PSNR metric. It is well-known that PSNR promotes blurry images, a limitation called the "regression to the mean" problem [61] in the SR litterature. Our image with the best PSNR comparison is the kriging component. We can explain it theoretically by the next Proposition 3 which shows that in expectation the best PSNR is provided by the mean of our samples, the kriging component.

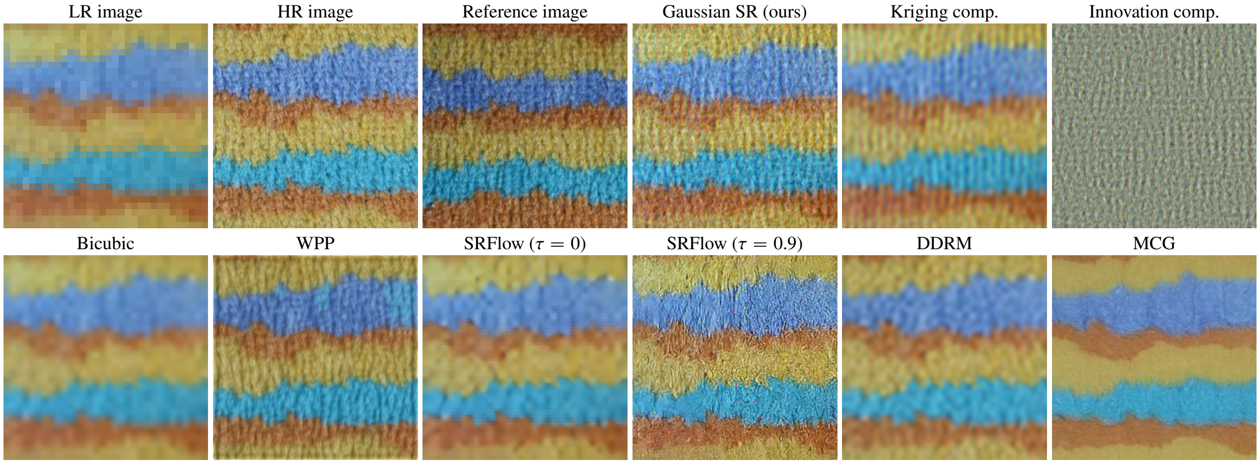

LR image
HR image
Reference image
Gaussian SR (ours)
Kriging comp.
Innovation comp.
Bicubic
WPP
SRFlow ($\tau = 0$)
SRFlow ($\tau = 0.9$)
DDRM
MCG

Figure 7: Comparison of our methods with stochastic SRFlow, MCG and DDRM and with the SISR WPP that uses the reference image (The HR size is 256×256 and $r = 8$.). Contrary to WPP, Gaussian SR is not influenced by the difference of luminance between the HR and the reference images.

Proposition 3 (Kriging component and MSE)**.** *Let* $\boldsymbol{u}_{\text{HR}} \in \mathbb{R}^{\Omega_{M,N}}$ *be a HR image,* $\boldsymbol{u}_{\text{LR}} = \boldsymbol{A}\boldsymbol{u}_{\text{HR}}$ *its LR version,* $\boldsymbol{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ *be the kriging operator and* $\boldsymbol{X}_{\text{SR}}$ *the random image following the distribution of the SR samples generated with Equation* (3) *then*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{X}_{\text{SR}}}\left(\|\boldsymbol{u}_{\text{HR}} - \boldsymbol{X}_{\text{SR}}\|_2^2\right) &= \|\boldsymbol{u}_{\text{HR}} - \boldsymbol{\Lambda}^T\boldsymbol{u}_{\text{LR}}\|_2^2 + \operatorname{Tr}\left[(\boldsymbol{I}_{\Omega_{M,N}} - \boldsymbol{\Lambda}^T\boldsymbol{A})\boldsymbol{\Gamma}(\boldsymbol{I}_{\Omega_{M,N}} - \boldsymbol{\Lambda}^T\boldsymbol{A})^T\right] \\ &\geqslant \|\boldsymbol{u}_{\text{HR}} - \boldsymbol{\Lambda}^T\boldsymbol{u}_{\text{LR}}\|_2^2.\end{aligned} \tag{12}$$

Simply put, the expected mean square error between the optimal HR image and Gaussian SR samples is always higher than the MSE between $\boldsymbol{u}_{\text{HR}}$ *and the associated kriging component* $\boldsymbol{\Lambda}^T\boldsymbol{u}_{\text{LR}}$*.*

Proposition 3, proved in Appendix D, means that adding the high frequency content of the innovation component is penalized by the PSNR since it is not aligned with the original high frequency content. Yet this addition is obviously perceptually important. We can also observe that SSIM has a similar behaviour as PSNR, as already observed for other degradation problems [30]. We argue that our study of Gaussian super resolution shows that LPIPS is the best metric to study the performance of stochastic SR algorithms.

Surprisingly, the three deep learning-based approaches give relatively poor results when applied to simple textures with high frequency content. This suggests that Imagenet and/or DIV2K datasets should probably be enriched with texture images to adapt the models for texture SR. Given its simplicity, computational efficiency, and qualitative superiority in comparison with state of the art SR methods, we believe our Gaussian SR algorithm is of interest for practitioners interested in stationary texture SR.

4.3 Study with measurement noise

Inverse problems are generally considered by modifying Equation (1) into

$$\boldsymbol{u}_{\text{LR}}^{\text{noisy}} = \boldsymbol{A}\boldsymbol{u}_{\text{HR}} + \sigma\mathbf{n}, \quad \sigma > 0, \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{13}$$

where $\sigma\mathbf{n}$ account for measurement noise. Theorems 1 and 2 still are valid and therefore we can apply a kriging reasoning, as detailed in Appendix E. It leads to introduce the kriging matrix $\boldsymbol{\Lambda}^{\text{noisy}} = (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$. Consequently, generating SR sample from this noisy observation can be achieved by applying the following equation:

$$\boldsymbol{u}_{\text{SR}}^{\text{noisy}} = \left[\boldsymbol{\Lambda}^{\text{noisy}}\right]^T\boldsymbol{u}_{\text{LR}}^{\text{noisy}} + \left[\tilde{\boldsymbol{u}} - \left[\boldsymbol{\Lambda}^{\text{noisy}}\right]^T(\boldsymbol{A}\tilde{\boldsymbol{u}} + \sigma\tilde{\mathbf{n}})\right], \quad \tilde{\boldsymbol{u}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \tilde{\mathbf{n}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{14}$$

Note that computing the kriging component $\left[\boldsymbol{\Lambda}^{\text{noisy}}\right]^T\boldsymbol{u}_{\text{LR}}^{\text{noisy}}$ can be interpreted as first apply a linear Wiener filter to the noisy image $\boldsymbol{u}_{\text{LR}}^{\text{noisy}}$ to obtain a denoised image $\mathbf{Wiener}(\boldsymbol{u}_{\text{LR}}^{\text{noisy}}) = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{u}_{\text{LR}}^{\text{noisy}}$ and then apply to

	$\sigma = 2/255$ (Figure 8)							
	Wall image				Fabric image			
	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓
Gaussian SR (Wiener)	18.93 ± 0.07	38.99 ± 0.08	0.12 ± 0.01	0.19 ± 0.01	19.62 ± 0.10	39.27 ± 0.06	0.19 ± 0.01	0.22 ± 0.01
Kriging comp. (Wiener)	21.15	37.29	0.19	0.75	20.84	37.52	**0.29**	0.53
WPP	19.97	38.77	0.13	0.22	19.29	24.07	0.23	**0.20**
DDRM	**21.52 ± 0.05**	**56.78 ± 0.00**	0.19 ± 0.00	0.69 ± 0.03	**22.34 ± 0.05**	**55.90 ± 0.00**	**0.29 ± 0.00**	0.55 ± 0.02
DPS	20.77 ± 0.52	20.77 ± 0.52	0.14 ± 0.01	0.61 ± 0.09	21.68 ± 0.17	29.80 ± 0.30	0.25 ± 0.01	0.55 ± 0.12
Gaussian SR (BM3D)	17.75 ± 0.11	35.39 ± 0.71	0.12 ± 0.00	**0.18 ± 0.01**	19.16 ± 0.07	40.54 ± 0.94	0.19 ± 0.00	0.23 ± 0.01
Kriging comp. (BM3D)	21.35	41.09	**0.20**	0.75	21.36	40.93	**0.29**	0.51

	$\sigma = 10/255$ (Figure 9)							
	Wall image				Fabric image			
	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓
Gaussian SR (Wiener)	17.30 ± 0.32	25.68 ± 0.28	0.08 ± 0.01	0.26 ± 0.01	19.13 ± 0.08	28.14 ± 0.14	0.17 ± 0.01	0.23 ± 0.01
Kriging comp. (Wiener)	17.66	25.31	0.15	0.76	20.11	27.76	0.26	0.64
WPP	19.38	25.89	0.11	0.23	17.50	18.58	0.20	**0.22**
DDRM	**21.15 ± 0.03**	**56.75 ± 0.00**	0.16 ± 0.00	0.66 ± 0.04	**21.86 ± 0.04**	**55.87 ± 0.00**	**0.27 ± 0.00**	0.60 ± 0.04
DPS	20.73 ± 0.34	20.73 ± 0.34	0.14 ± 0.01	0.61 ± 0.09	21.58 ± 0.22	27.91 ± 0.14	0.25 ± 0.01	0.57 ± 0.11
Gaussian SR (BM3D)	17.68 ± 0.06	32.11 ± 0.17	0.10 ± 0.00	**0.20 ± 0.01**	18.56 ± 0.05	27.48 ± 0.06	0.17 ± 0.00	0.25 ± 0.01
Kriging comp. (BM3D)	21.11	32.70	**0.17**	0.67	21.09	30.34	**0.27**	0.58

Table 3: Quantitative comparison of our Gaussian SR methods with other algorithms in the presence of noise at low or high level ($\sigma = 2/255$ or $\sigma = 10/255$). The corresponding images are in Figures 8 and 9. Note that the texture information is again promoted by the LPIPS metric.

this denoised image a previous kriging matrix of the noiseless case $\boldsymbol{\Lambda}^T = \boldsymbol{\Gamma}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T)^\dagger$. This is proved in detail in Appendix E. Consequently, Equation (14) can be rewritten as

$$\boldsymbol{u}_{\mathrm{SR}}^{\mathrm{Wiener}} = \boldsymbol{\Lambda}^T\, \mathbf{Wiener}(\boldsymbol{u}_{\mathrm{LR}}^{\mathrm{noisy}}) + \left[\tilde{\boldsymbol{u}} - \boldsymbol{\Lambda}^T\, \mathbf{Wiener}(\boldsymbol{A}\tilde{\boldsymbol{u}} + \sigma\tilde{\mathbf{n}})\right], \tilde{\boldsymbol{u}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \tilde{\mathbf{n}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{15}$$

A natural improvement of the algorithm is to replace the linear Wiener filter by a more efficient denoising algorithm. We illustrate the results obtained with the patch-based BM3D algorithm [10] (using the online version [42]). The equation verified by the samples becomes

$$\boldsymbol{u}_{\mathrm{SR}}^{\mathrm{BM3D}} = \boldsymbol{\Lambda}^T\, \mathbf{BM3D}(\boldsymbol{u}_{\mathrm{LR}}^{\mathrm{noisy}}) + \left[\tilde{\boldsymbol{u}} - \boldsymbol{\Lambda}^T\, \mathbf{BM3D}(\boldsymbol{A}\tilde{\boldsymbol{u}} + \sigma\tilde{\mathbf{n}})\right], \tilde{\boldsymbol{u}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \tilde{\mathbf{n}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{16}$$

where $\boldsymbol{\Lambda}$ verifies the noiseless kriging Equation (2) and $\mathbf{BM3D}(\boldsymbol{u}_{\mathrm{LR}}^{\mathrm{noisy}})$ is the output of BM3D applied to $\boldsymbol{u}_{\mathrm{LR}}^{\mathrm{noisy}}$. More modern algorithms could be considered such as the Denoising Convolutional Neural Network (DnCNN) [70] or Denoising Residual UNet (DRUNet) [69] but BM3D is learning-free and provides already good results.

We compare our two methods “Gaussian SR (Wiener)” and “Gaussian SR (BM3D)” with DDRM and WPP that can both be adapted to this noisy framework. We present also Diffusion Posterior Sampling (DPS[6]) [8] which is a version of MCG adapted to the noisy setting, without projection on the image space of $\boldsymbol{A}$. Results are summarized in Figures 8 and 9 and table 3. A weak level of noise $\sigma = 2/255$, corresponding to a quantification degradation, does not harm our method performance. We note that BM3D provides a slightly sharper samples than the Wiener filter. However, “Gaussian SR (Wiener)” fails with a higher amount of noise ($\sigma = 10/255$). The components contain color artifacts as shown in Figure 9, which results from the limitation of the linear Wiener filter to denoise the noisy LR image. “Gaussian SR (BM3D)” outputs more realistic samples but may contain frequency artifact, as illustrated by the kriging component of the wall texture. Promoted by the LPIPS metric (see Table 3), Gaussian SR still provides relevant texture restoration. WPP is also competitive in term of LPIPS metric and is the best method in presence of a higher level of noise. Unsurprisingly, Gaussian SR fails to respect the constraint $\boldsymbol{u}_{\mathrm{LR}} = \boldsymbol{A}\boldsymbol{u}_{\mathrm{HR}}$, illustrated by the LR-PSNR metric (using the unknown LR image). In this noisy case, this metric reports on the quality of the denoising. The other deep learning methods are more robust and adapted to this noisy case.

4.4 Extension to other degradation operators

Our method can be applied to reconstruct images degraded by other degradation operators in the form $\boldsymbol{SC}$ where $\boldsymbol{C}$ is a convolution preserving the mean. All the proofs and the properties still hold. To illustrate it, we consider

---

[6] Code and weights from `https://github.com/DPS2022/diffusion-posterior-sampling`

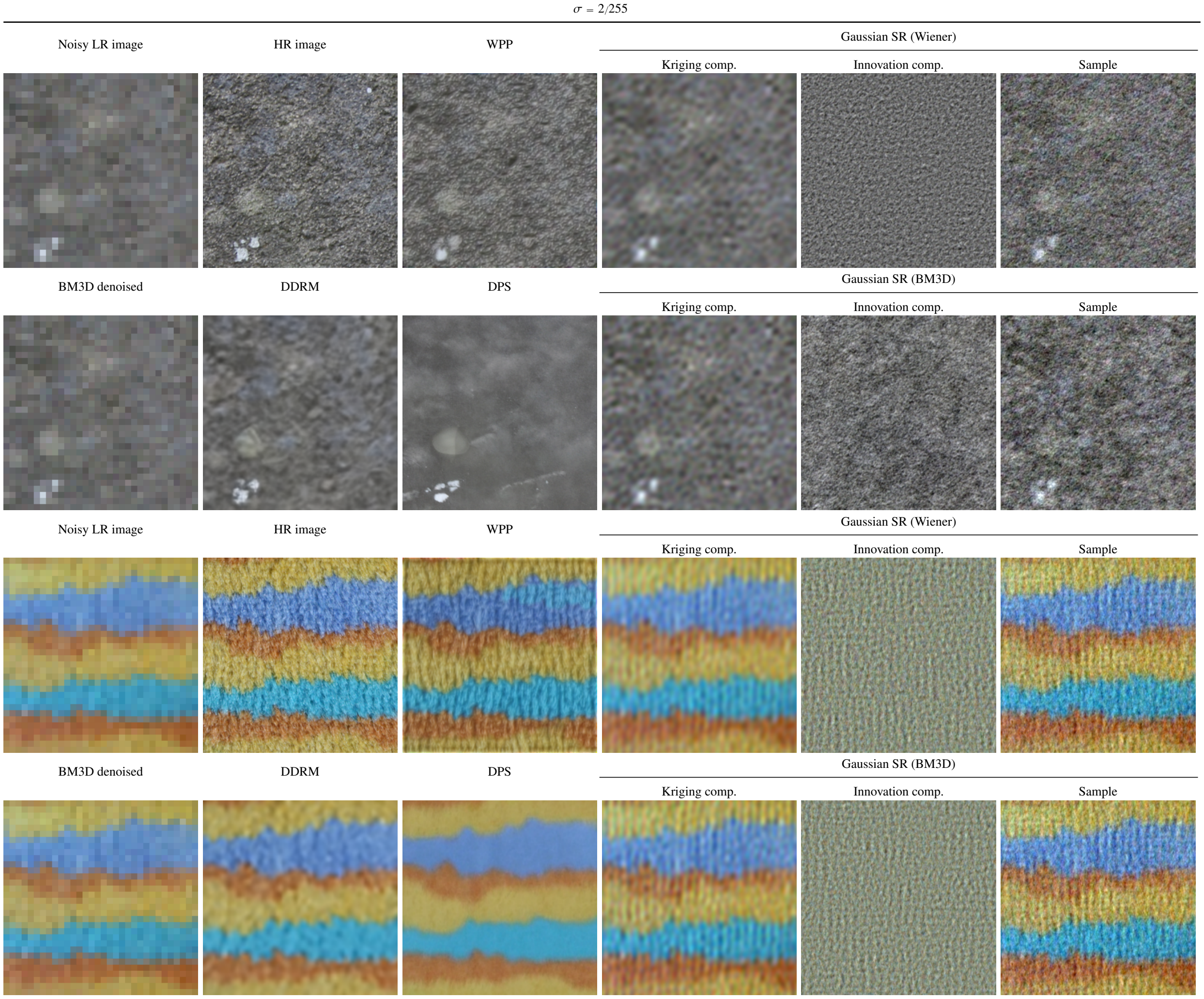


Figure 8: Comparison of our algorithms (“Gaussian SR (Wiener)” and “Gaussian SR (BM3D)”) in the presence of a low level of noise ($\sigma = 2/255$) with the DPS, the WPP and the DDRM methods. HR and reference image are identical to Figures 6 and 7. Note that a weak amount of noise does not affect our results.

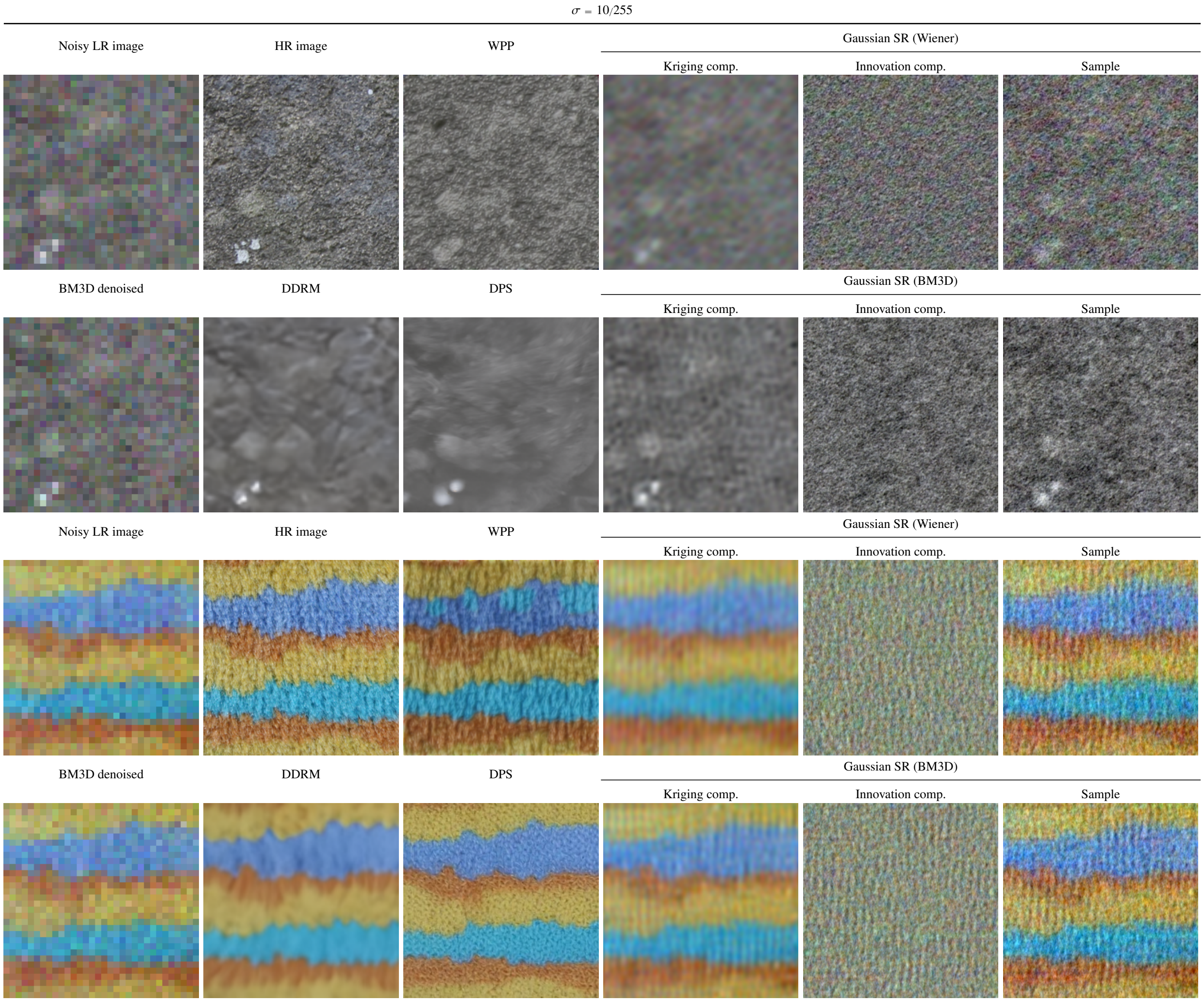
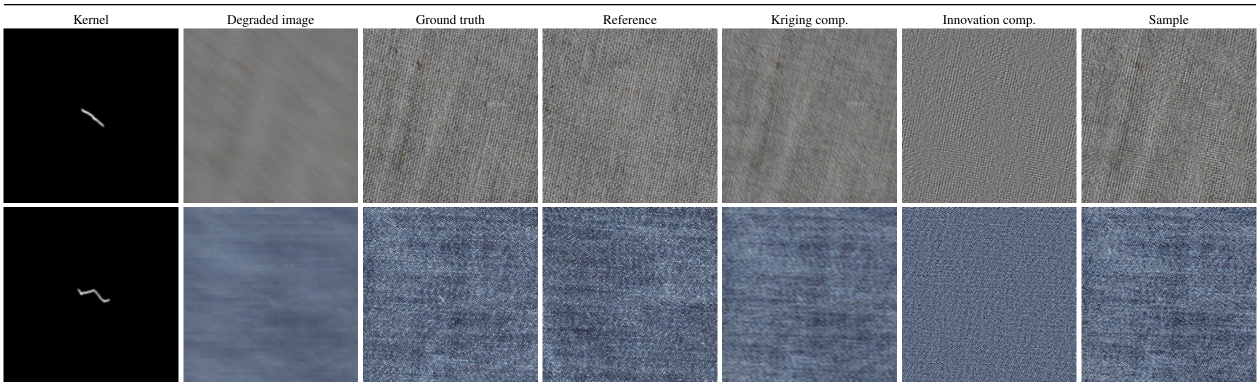


Figure 9: Comparison of our algorithms ("Gaussian SR (Wiener)" and "Gaussian SR (BM3D)") in the presence of a high level of noise ($\sigma = 10/255$) with the DPS, the WPP and the DDRM methods. HR and reference image are identical to Figures 6 and 7. Note that a high amount of noise creates color artefacts in our results.

Motion blur followed by a subsampling operator with stride $r = 4$



Motion blur followed by a bicubic convolution and a subsampling operator with stride $r = 4$

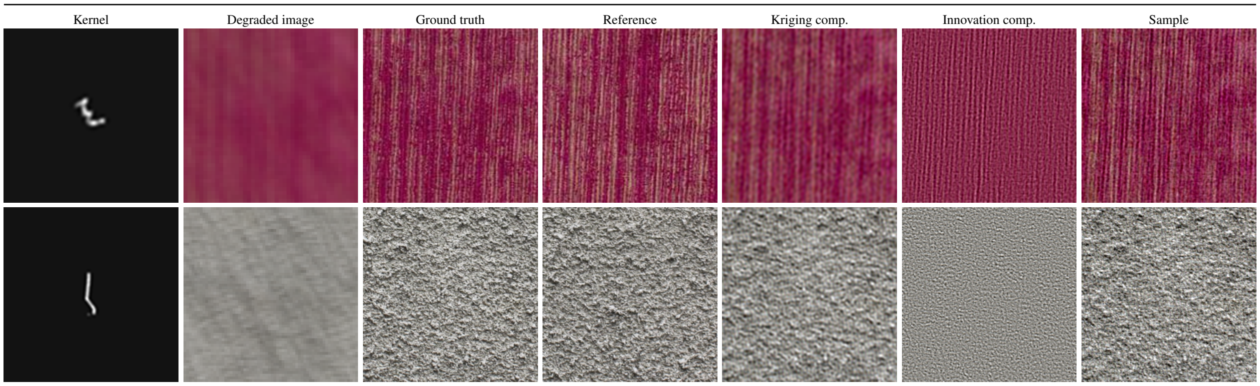


Figure 10: Illustration of the application of Gaussian kriging to other linear operators. The images have square size which are respectively $208, 208, 176$ and 256. The two first images are degraded by a motion blur, followed by a subsampling with stride $r = 4$ and the two others are degraded by a motion blur followed by the zoom-out bicubic operator with factor $r = 4$.

motion blurs randomly generated by the code[7] with kernel size 61×61 and intensity value 0.5, following [8]. We propose to reconstruct images degraded by a motion blur followed by a subsampling operator $\boldsymbol{S}$ with stride $r = 4$ and images degraded by a motion blur and the bicubic zoom-out operator $\boldsymbol{A}$ with factor $r = 4$. These are not standard problems but they can be seen as multiple problems as studied in [59]. In Figure 10, we present the results provided by our method. We can again observe the complementarity of the deterministic kriging component and the stochastic innovation component that retrieves the perceptual texture grain.

5 Limitations

As shown in the previous section, when applied to Gaussian textures our Gaussian SR algorithm outperforms state-of-the-art methods regarding results quality and execution time. However, Gaussian SR has several inherent limitations that we now highlight.

5.1 Limited scope

Gaussian microtextures represent a limited scope of images. The images are supposed to be stationary with no geometric structures, excluding a lot of textures such as the brick wall in Figure 11. When applying our Gaussian SR algorithm to such a structured texture, one can observe that the independent stationary stochastic innovation component is not aligned with the brick lines recovered in the kriging component. Also, textures with too big individual objects are unadapted to the Gaussian microtexture assumption, such as the bark texture in the second row of Figure 11. The

---

[7] `https://github.com/LeviBorodenko/motionblur`

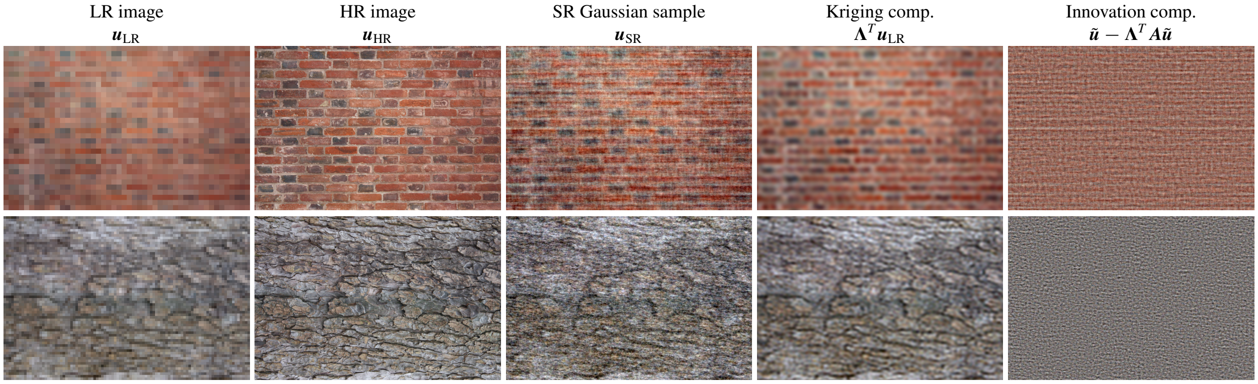


Figure 11: Examples of the application of Gaussian SR on images that are not Gaussian microtextures (The HR size is 512×768 and $r = 16$). The first image has a too structured texture to be treated as a Gaussian texture. The second image cannot be considered as a microtexture due to the presence of too big individual patterns.

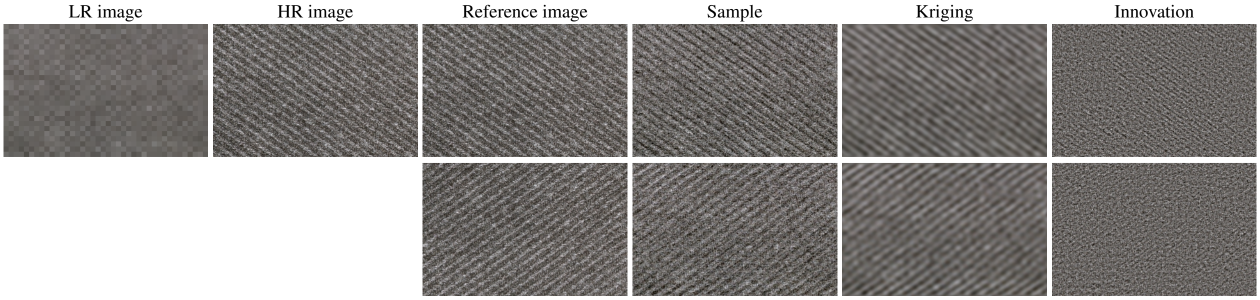


Figure 12: Illustration of the importance of the choice of the reference image. Each row presents the samples of Gaussian SR obtained with the same LR image and a different reference image (The HR size is 208×320 and $r = 8$.). The covariance information provided by the reference image guides the final texture sample.

kriging component is attached to the LR data but the innovation component provides a stationary grainy texture that is not in accordance with the piecewise flat nature of the texture. This is a well-known limitation of the ADSN model [22].

5.2 Non-adapted reference image

The choice of the reference image has a strong impact on the final samples. The texture is essentially provided by the innovation component which is built with ADSN($\boldsymbol{u}_{\mathrm{ref}}$). Figure 12 presents two different samples we can obtain depending on the choice of the reference image. The first reference image is the perfect HR image that retrieves the correct alignment of the texture lines. On the contrary, a reference image with opposite alignment modifies the texture of the Gaussian samples.

More generally, with a reference image, the stability result of Proposition 2 does not hold for the simulation of the kriging component. Algorithm 1 can lead to instabilities as illustrated in Figure 13. Here we observe that a frequency has been intensified in the kriging component and provokes artificial parallel lines in the final sample. It is essentially due to a slight misalignment between the HR texture and the reference texture. Note that there is no artefact in the innovation component which fits the assumption of Proposition 2. This type of observations is marginal but can occur in some rare cases.

5.3 Non-adaptative variance

We can study the variance of the samples generated by our method with Proposition 4. As a reminder, in our approximation $\boldsymbol{\Lambda}_{\mathrm{approx}}^T$ is in the form $\boldsymbol{\Gamma}\boldsymbol{A}^T\boldsymbol{C}_{\boldsymbol{\eta}}$ where $\boldsymbol{C}_{\boldsymbol{\eta}}$ is a diagonal multi-chanel convolution but this is also the case with the perfect solution of Equation (2) $\boldsymbol{\Lambda} = \boldsymbol{\Gamma}\boldsymbol{A}^T\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^{\dagger}$. This proposition shows that the images sampled with Gaussian

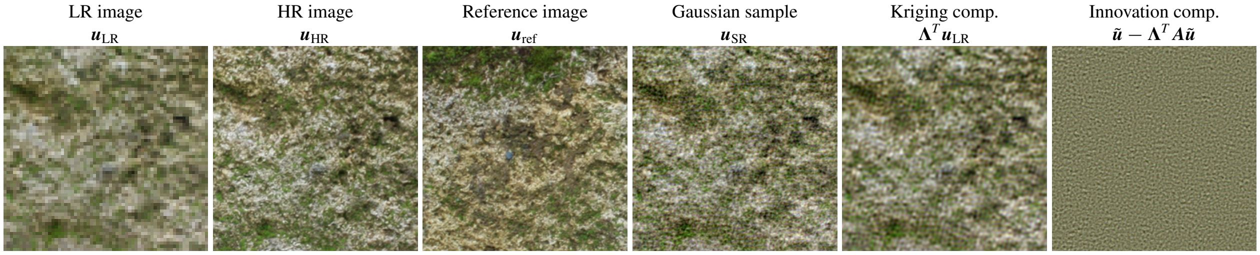


Figure 13: Example of Gaussian SR that illustrates the non-application of Proposition 2 in the setting with reference image (HR size is 256×256 and $r = 4$). A frequency has been amplified and provokes parallel lines in the kriging component generated by Gaussian SR. This artefact occurs in rare cases. This phenomenon is not observable in the innovation component which verifies the assumptions of Proposition 2.

SR have a constant variance on sub-grids of the image domain. The law of the SR samples $\boldsymbol{u}_{\mathrm{SR}}$ inherits the stationarity of $\mathrm{ADSN}(\boldsymbol{u}_{\mathrm{ref}})$ being invariant by the translations by $(kr, \ell r)$ where $k, \ell \in \mathbb{Z}$. In other words, the law of our SR samples is a cyclostationary law [47].

Proposition 4 (Non-adaptative variance of Gaussian SR)**.** *Let* $\boldsymbol{\Lambda}^T \in \mathbb{R}^{3\Omega_{M,N} \times 3\Omega_{M/r,N/r}}$ *being in the form* $\boldsymbol{C}_{\boldsymbol{\eta}}\boldsymbol{S}^T$ *where the kernels* $\boldsymbol{\eta} \in \mathbb{R}^{3\Omega_{M,N}}$*. Then, the law of the SR samples generated by Equation* (3) *is invariant by translations by* $(kr, \ell r)$ *where* $k, \ell \in \mathbb{Z}$*.*

Proposition 4 implies that the samples generated by Gaussian SR have a pixelwise variance constant by translations $(kr, \ell r)$ where $k, \ell \in \mathbb{Z}$. This variance can be computed theoretically and compared to the empirical pixelwise variance of the other stochastic algorithms, as done in Figure 14. The white spot detail in the wall texture is taking into in account in the variety of the samples proposed by the other stochastic super-resolution algorithms but not by our Gaussian SR. The variance of SRFlow is quite high and allow us to see the input image, it is clearly adapted to the LR image. MCG variance maps show the details of the images but one can guess that the texture is not really retrieved because the variance is smooth on the rest of the image. Besides, DDRM is almost deterministic with a low pixel variance value (see also the variation of the metrics in Table 2). Gaussian SR presents a constant variance on sub-grids of the image. Indeed, one of the main limitation of the model is that the innovation component is independent of the kriging component.

6 Conclusion

This work provides an efficient sampling algorithm for solutions of the stochastic SR problem for Gaussian microtextures. Our solution is exact for grayscale images and a we propose a fast and reliable approximation for RGB images using a well-known mathematical kriging reasoning. Gaussian SR is a fast and qualitatively better than involved generic methods when applied to Gaussian microtextures accompanied with a reference HR image. We have demonstrated that it can be extended to noisy observations using the natural Wiener filter associated with the Gaussian model. This Wiener filter can be replaced by a more complex denoiser such as BM3D for more robust performance. Besides, of a more general interest, our comparative study shows that LPIPS is a pertinent metric to evaluate stochastic SR results that do not suffer from excessive blur. While our Gaussian SR has inherent limitations discussed in the paper, it could be of interest for fast and reliable restoration of HR stationary microtextures, even in the presence of motion blur.

Acknowledgments

The authors acknowledge the support of the MISTIC project (ANR-19-CE40-005) and the CaSciModOT federation (Calcul Scientifique et Modélisation Orléans-Tours). We also wish to thank the reviewers for their insightful comments, which significantly contributed to improving the quality of the manuscript.

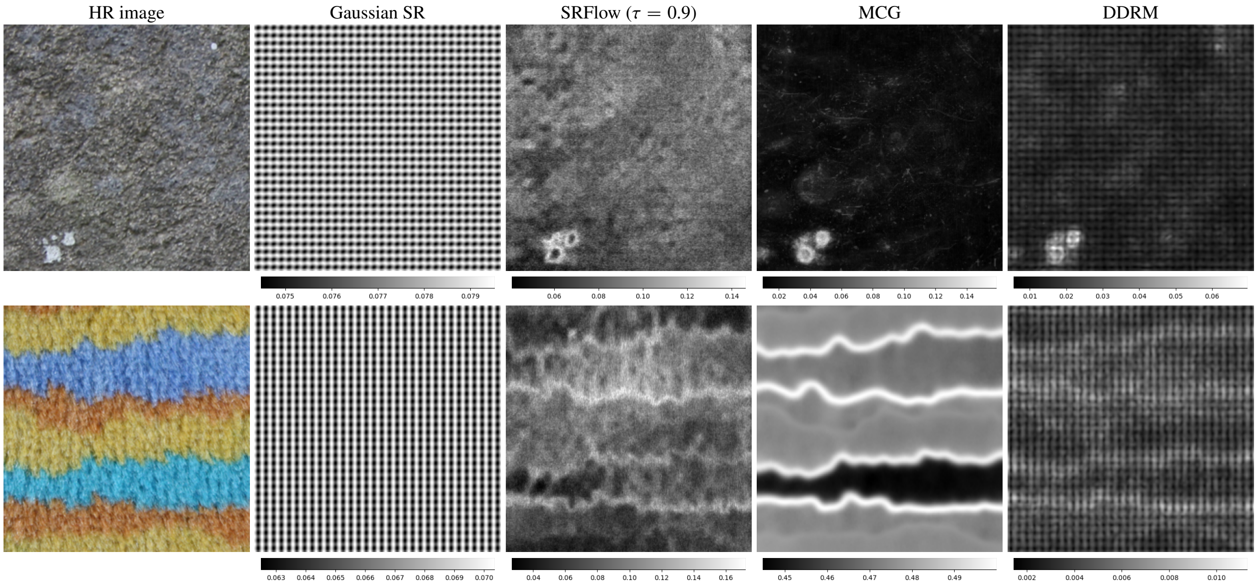


Figure 14: Standard deviation maps of the different algorithms for stochastic super-resolution applied to the HR images of Figures 6 and 7, with $r = 8$. The variance of Gaussian SR has been computed theoretically while it has been estimated with 100 samples for the others, and the sum of the pixelwise variance over the three chanels is displayed. The deep learning approaches DDRM, SRFlow and DPS have an adaptative variance to the HR image, showing the details of the images. In comparison, the Gaussian SR has a cyclostationary non-adaptative variance. The range of the images is $[0, 1]$.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] Fabian Altekrüger and Johannes Hertrich. WPPNets and WPPFlows: The power of Wasserstein patch priors for superresolution. *SIAM Journal on Imaging Sciences*, 16(3):1033–1067, 2023.

[3] Fabian Altekrüger, Alexander Denker, Paul Hagemann, Johannes Hertrich, Peter Maass, and Gabriele Steidl. Patchnr: learning from very few images by patch normalizing flow regularization. *Inverse Problems*, 39(6):064006, may 2023.

[4] Yuval Bahat and Tomer Michaeli. Explorable super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2716–2725, 2020.

[5] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[6] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004.

[7] Pierrick Chatillon, Yann Gousseau, and Sidonie Lefebvre. A statistically constrained internal method for single image super-resolution. In *26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022*, pages 1322–1328. IEEE, 2022.

[8] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

[9] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. 2022.

[10] K. . Dabov, A. . Foi, V. . Katkovnik, and K. . Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Trans. Img. Proc.*, 16(8):2080–2095, August 2007.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] A. Desolneux, L. Moisan, and S. Ronsin. A compact representation of random phase and Gaussian textures. In *ICASSP'12*, pages 1381–1384, 2012.

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

[14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, 2014.

[15] J.L. Doob. *Stochastic Processes*. Probability and Statistics Series. Wiley, 1953.

[16] Gilad Freedman and Raanan Fattal. Image and video upscaling from local self-examples. *ACM Trans. Graph.*, 28(3):1–10, 2010.

[17] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[18] B. Galerne, A. Lagae, S. Lefebvre, and G. Drettakis. Gabor noise by example. *ACM Trans. Graph.*, 31(4):73:1–73:9, jul 2012.

[19] B. Galerne, A. Leclaire, and L. Moisan. A texton for fast and flexible Gaussian texture synthesis. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pages 1686–1690, 2014.

[20] B. Galerne, A. Leclaire, and L. Moisan. Texton noise. *Computer Graphics Forum*, 2017.

[21] B. Galerne, A. Leclaire, and J. Rabin. A texture synthesis model based on semi-discrete optimal transport in patch space. *SIAM Journal on Imaging Sciences*, 11(4):2456–2493, 2018.

[22] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, 20(1):257–267, 2011.

[23] Bruno Galerne and Arthur Leclaire. An Algorithm for Gaussian Texture Inpainting. *Image Processing On Line*, 7:262–277, 2017. `https://doi.org/10.5201/ipol.2017.198`.

[24] Bruno Galerne and Arthur Leclaire. Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*, 10(3):1446–1474, 2017.

[25] Bruno Galerne, Arthur Leclaire, and Lionel Moisan. Microtexture inpainting through Gaussian conditional simulation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1204–1208, 2016.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[27] J. Gutierrez, B. Galerne, J. Rabin, and T. Hurtut. Optimal patch assignment for statistically constrained texture synthesis. In F. Lauze, Y. Dong, and A. B. Dahl, editors, *Scale Space and Variational Methods in Computer Vision: 6th International Conference, SSVM 2017, Kolding, Denmark, June 4-8, 2017, Proceedings*, pages 172–183, Cham, 2017. Springer International Publishing.

[28] Johannes Hertrich, Antoine Houdard, and Claudia Redenbach. Wasserstein Patch Prior for Image Superresolution. *IEEE Transactions on Computational Imaging*, 8:693–704, 2022.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[30] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[31] Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, and Julien Rabin. Wasserstein generative models for patch-based texture synthesis. In Abderrahim Elmoataz, Jalal Fadili, Yvain Quéau, Julien Rabin, and Loïc Simon, editors, *Scale Space and Variational Methods in Computer Vision*, pages 269–280, Cham, 2021. Springer International Publishing.

[32] Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, and Julien Rabin. A generative model for texture synthesis based on optimal transport between feature distributions. *J. Math. Imaging Vis.*, 65(1):4–28, 2023.

[33] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang. Sampling network guided cross-entropy method for unsupervised point cloud registration. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6108–6117, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society.

[34] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[35] Ulugbek S. Kamilov, Charles A. Bouman, Gregery T. Buzzard, and Brendt Wohlberg. Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97, 2023.

[36] W. J. Kammerer and M. Z. Nashed. On the convergence of the conjugate gradient method for singular linear operator equations. *SIAM Journal on Numerical Analysis*, 9(1):165–181, 1972.

[37] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[38] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016.

[39] Durk P Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[40] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5835–5843, 2017.

[41] C. Lantuejoul. *Geostatistical Simulation: Models and Algorithms*. Collezione legale Pirola. Springer Berlin Heidelberg, 2001.

[42] Marc Lebrun. An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line*, 2:175–213, 2012. `https://doi.org/10.5201/ipol.2012.l-bm3d`.

[43] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017.

[44] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[45] Z. Liu, W. Siu, and L. Wang. Variational autoencoder for reference based image super-resolution. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 516–525, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.

[46] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. SRFlow: Learning the Super-Resolution Space with Normalizing Flow. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, pages 715–732, Berlin, Heidelberg, 2020. Springer-Verlag. event-place: Glasgow, United Kingdom.

[47] N. Lutz, B. Sauvage, and J-M. Dischler. Cyclostationary gaussian noise: theory and synthesis. *Computer Graphics Forum*, 40:xx–yy, 2021.

[48] Simon Mignon, Bruno Galerne, Moncef Hidane, Cécile Louchet, and Julien Mille. Semi-unbalanced regularized optimal transport for image restoration. In *31st European Signal Processing Conference, EUSIPCO 2023, Helsinki, Finland, September 4-8, 2023*, pages 466–470. IEEE, 2023.

[49] Simon Mignon, Bruno Galerne, Moncef Hidane, Cécile Louchet, and Julien Mille. Semi-unbalanced optimal transport for image restoration and synthesis. working paper or preprint, 2024.

[50] Lionel Moisan. Periodic Plus Smooth Image Decomposition. *Journal of Mathematical Imaging and Vision*, 39(2):161–179, 2011.

[51] Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[52] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.

[53] Émile Pierret and Bruno Galerne. Stochastic super-resolution for gaussian textures. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[54] Jean Prost, Antoine Houdard, Andrés Almansa, and Nicolas Papadakis. Learning local regularization for variational image restoration. In *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Virtual Event, May 16–20, 2021, Proceedings*, page 358–370, Berlin, Heidelberg, 2021. Springer-Verlag.

[55] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[56] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution Via Iterative Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2022.

[57] Mehdi S. M. Sajjadi, Bernhard Schölkopf, and Michael Hirsch. EnhanceNet: Single Image Super-Resolution through Automated Texture Synthesis. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4501–4510. IEEE, 2017.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[59] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations (ICLR)*, May 2023.

[60] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. 2023.

[61] C. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. In *International Conference on Learning Representations*, 2017.

[62] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[63] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. *Int. J. Comput. Vis.*, 128(7):1867–1888, 2020.

[64] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[65] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 63–79, Cham, 2019. Springer International Publishing.

[66] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[67] G.-S. Xia, S. Ferradans, G. Peyré, and J.-F. Aujol. Synthesizing and mixing stationary Gaussian texture models. *SIAM J. on Imaging Science*, 8(1):476–508, 2014.

[68] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *Image Processing, IEEE Transactions on*, 21(5):2481–2499, May 2012.

[69] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

[70] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[71] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, Los Alamitos, CA, USA, June 2018. IEEE Computer Society.

[72] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 87–104, Cham, 2018. Springer International Publishing.

[73] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486, 2011.

A Proof of Theorem 2

The fact that the conditional expectation is linear for zero-mean Gaussian multivariate law is a classical result (see e.g. [15]). Let us denote $(\boldsymbol{\lambda}(x))_{x\in\Omega_{M,N}}$ the columns of $\boldsymbol{\Lambda}$. $\boldsymbol{\Lambda}$ minimizes

$$\begin{aligned}\mathbb{E}\left(\left\|\boldsymbol{\Lambda}^T\boldsymbol{A}\boldsymbol{X}-\boldsymbol{X}\right\|_2^2\right) &= \mathbb{E}\left[\sum_{x\in\Omega_{M,N}}\left((\boldsymbol{\Lambda}^T\boldsymbol{A}\boldsymbol{X})(x)-\boldsymbol{X}(x)\right)^2\right]\\ &= \sum_{x\in\Omega_{M,N}}\mathbb{E}\left[\left(\boldsymbol{\lambda}(x)^T\boldsymbol{A}\boldsymbol{X}-\boldsymbol{X}(x)\right)^2\right].\end{aligned}$$

This is a separable function of the columns $(\boldsymbol{\lambda}(x))_{x\in\Omega_{M,N}}$. For each $x \in \Omega_{M,N}$, let us minimize $\boldsymbol{\lambda}(x) \in \mathbb{R}^{\Omega_{M/r,N/r}} \mapsto \mathbb{E}\Big[\big(\boldsymbol{\lambda}(x)^T\boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}(x)\big)^2$
One has

$$\begin{aligned}\mathbb{E}\left[\left(\boldsymbol{\lambda}(x)^T\boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}(x)\right)^2\right] &= \mathrm{Var}(\boldsymbol{\lambda}(x)^T\boldsymbol{A}\boldsymbol{X}) + \mathrm{Var}(\boldsymbol{X}(x)) - 2\,\mathrm{Cov}(\boldsymbol{\lambda}(x)^T\boldsymbol{A}\boldsymbol{X}, \boldsymbol{X}(x))\\ &= \boldsymbol{\lambda}(x)^T\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\boldsymbol{\lambda}(x) + \mathrm{Var}(\boldsymbol{X}(x)) - 2\boldsymbol{\lambda}(x)^T\boldsymbol{A}\,\mathrm{Cov}(\boldsymbol{X}, \boldsymbol{X}(x)).\end{aligned}$$

This is a quadratic functional associated with the positive matrix $2\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T$ and the vector $2\boldsymbol{A}\,\mathrm{Cov}(\boldsymbol{X}, \boldsymbol{X}(x)) = 2\boldsymbol{A}\boldsymbol{\Gamma}_{\Omega_{M,N}\times\{x\}}$. It is minimal for $\boldsymbol{\lambda}(x)$ any solution of the linear system $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\boldsymbol{\lambda}(x) = \boldsymbol{A}\boldsymbol{\Gamma}_{\Omega_{M,N}\times\{x\}}$. This is valid for any column $\boldsymbol{\lambda}(x)$ of $\boldsymbol{\Lambda}$. As a consequence, $\boldsymbol{\Lambda}$ is a solution if and only if it verifies

$$\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\boldsymbol{\Lambda} = \boldsymbol{A}\boldsymbol{\Gamma}.$$

B Proof of Lemma 1

It is sufficient to identify the convolutions on the canonical basis. For $x \in \Omega_{M,N}$, we define $\delta_x \in \mathbb{R}^{\Omega_{M,N}}$ such that for $z \in \Omega_{M,N}$, $\delta_x(z) = \mathbb{1}_{x=z}$ and identically δ^r_x for $x \in \Omega_{M/r,N/r}$. Denoting T_x the translation by $x \in \Omega_{M,N}$, that is for all $z \in \mathbb{R}^{\Omega_{M,N}}$, $\mathrm{T}_x\boldsymbol{u}(z) = \boldsymbol{u}(z-x)$. Remark that convolving with $\delta_x^{\Omega_{M,N}}$ corresponds to translating by x: for all $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}, \boldsymbol{u} \star \delta_x = \mathrm{T}_x\boldsymbol{u}$.

1. For $x \in \Omega_{M/r,N/r}$,
$$\boldsymbol{S}\boldsymbol{C}_{\boldsymbol{\alpha}}\boldsymbol{S}^T\delta^r_x = \boldsymbol{S}\boldsymbol{C}_{\boldsymbol{\alpha}}\delta_{rx} = \boldsymbol{S}(\boldsymbol{\alpha}\star\delta_{rx}) = \boldsymbol{S}\mathrm{T}_{rx}\boldsymbol{\alpha} = \mathrm{T}_x\boldsymbol{S}\boldsymbol{\alpha} = (\boldsymbol{S}\boldsymbol{\alpha})\star\delta^r_x. \tag{17}$$

2. For $x \in \Omega_{M/r,N/r}$,
$$\boldsymbol{S}^T\boldsymbol{C}_{\boldsymbol{\beta}}\delta^r_x = \boldsymbol{S}^T(\boldsymbol{\beta}\star\delta^r_x) = \boldsymbol{S}^T\mathrm{T}_x\boldsymbol{\beta} = \mathrm{T}_{rx}\boldsymbol{S}^T\boldsymbol{\beta} = (\boldsymbol{S}^T\boldsymbol{\beta})\star\delta_{rx} = (\boldsymbol{S}^T\boldsymbol{\beta})\star(\boldsymbol{S}^T\delta^r_x). \tag{18}$$

Finally, the equality of the convolutions on the canonical basis is established.

C Conjugate Gradient Descent algorithm

The CGD algorithm is an iterative method to approximate the solution of a linear equation by solving the associated least-squares problem [36]. Algorithm 2 aims at solving the normal equation

$$\boldsymbol{B}^T\boldsymbol{B}\psi = \boldsymbol{B}^T\varphi \tag{19}$$

associated with the least-squares problem $\psi \mapsto \|\boldsymbol{B}\psi - \varphi\|_2$ and has good convergence properties when applied to the inpainting problem for Gaussian textures [24]. First we observed that this also applies in the case of super-resolution. In our SR context, our goal is to express $\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T\right)^\dagger\boldsymbol{v}$ for a given LR image $\boldsymbol{v} \in \mathbb{R}^{\Omega_{M/r,N/r}}$. It is equivalent to minimize $\psi \in \mathbb{R}^{\Omega_{M/r,N/r}} \mapsto \left\|(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A})^T\psi - \boldsymbol{v}\right\|_2$. Therefore, we apply CGD with $B = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T$ and $\varphi = \boldsymbol{v}$.

Algorithm 2 CGD Algorithm CGD to compute $\boldsymbol{B}^\dagger\varphi$

```
Input: Initialize $k \leftarrow 0, \psi_0 \leftarrow 0, r_0 \leftarrow \boldsymbol{B}^T\varphi - \boldsymbol{B}^T\boldsymbol{B}\psi_0, d_0 \leftarrow r_0$
while $\|r_k\|_2 > \varepsilon$ do
    $\alpha_k \leftarrow \frac{\|r_k\|_2^2}{d_k^T\boldsymbol{B}^T\boldsymbol{B}d_k}$
    $\psi_{k+1} \leftarrow \psi_k + \alpha_k d_k$
    $r_{k+1} \leftarrow r_k - \alpha_k\boldsymbol{B}^T\boldsymbol{B}d_k$
    $d_{k+1} \leftarrow r_{k+1} + \frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2}d_k$
    $k \leftarrow k + 1$
end while
Output: $\psi_k$
```

D Proof of Proposition 3

Let $\boldsymbol{u}_{\text{HR}} \in \mathbb{R}^{\Omega_{M,N}}$ be a HR image, $\boldsymbol{u}_{\text{LR}} = \boldsymbol{A}\boldsymbol{u}_{\text{HR}}$ its LR version, $\boldsymbol{u}_{\text{ref}}$ a reference image, $\boldsymbol{\Gamma}$ such that $\text{ADSN}(\boldsymbol{u}_{\text{ref}}) = \mathscr{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$, $\boldsymbol{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ be the kriging operator and $\boldsymbol{X}_{\text{SR}}$ the random image following the distribution of the SR samples generated with Equation (3). $\boldsymbol{X}_{\text{SR}}$ has the same law as:

$$\boldsymbol{u}_{\text{SR}} = \boldsymbol{\Lambda}^T \boldsymbol{u}_{\text{LR}} + \tilde{\boldsymbol{X}} - \boldsymbol{\Lambda}^T \boldsymbol{A} \tilde{\boldsymbol{X}}$$

with $\tilde{\boldsymbol{X}} \sim \mathscr{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$. Consequently, considering $\mathbb{E}_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{X}}) = \boldsymbol{0}$,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{X}_{\text{SR}}}\left(\|\boldsymbol{u}_{\text{HR}} - \boldsymbol{X}_{\text{SR}}\|_2^2\right) =& \|\boldsymbol{u}_{\text{HR}} - \boldsymbol{\Lambda}^T \boldsymbol{u}_{\text{LR}}\|_2^2 + \mathbb{E}_{\tilde{\boldsymbol{X}}}\left(\|\tilde{\boldsymbol{X}} - \boldsymbol{\Lambda}^T \boldsymbol{A} \tilde{\boldsymbol{X}}\|_2^2\right) \\
&+ 2\langle \boldsymbol{u}_{\text{HR}} - \boldsymbol{\Lambda}^T \boldsymbol{u}_{\text{LR}}, \mathbb{E}_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{X}}) - \boldsymbol{\Lambda}^T \boldsymbol{A} \mathbb{E}_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{X}}) \rangle \\
=& \|\boldsymbol{u}_{\text{HR}} - \boldsymbol{\Lambda}^T \boldsymbol{u}_{\text{LR}}\|_2^2 + \text{Tr}\left[(\boldsymbol{I}_{\Omega_{M,N}} - \boldsymbol{\Lambda}^T \boldsymbol{A}) \boldsymbol{\Gamma} (\boldsymbol{I}_{\Omega_{M,N}} - \boldsymbol{\Lambda}^T \boldsymbol{A})^T\right]
\end{aligned}$$

And, $(\boldsymbol{I}_{\Omega_{M,N}} - \boldsymbol{\Lambda}^T \boldsymbol{A}) \boldsymbol{\Gamma} (\boldsymbol{I}_{\Omega_{M,N}} - \boldsymbol{\Lambda}^T \boldsymbol{A})^T$ is a positive semi-definite matrix with non-negative trace which gives

$$\mathbb{E}_{\boldsymbol{X}_{\text{SR}}}\left(\|\boldsymbol{u}_{\text{HR}} - \boldsymbol{X}_{\text{SR}}\|_2^2\right) \geqslant \|\boldsymbol{u}_{\text{HR}} - \boldsymbol{\Lambda}^T \boldsymbol{u}_{\text{LR}}\|_2^2. \tag{20}$$

E Study of the noisy case

Considering the equation

$$\boldsymbol{u}_{\text{LR}}^{\text{noisy}} = \boldsymbol{A}\boldsymbol{u}_{\text{HR}} + \sigma \mathbf{n}, \quad \mathbf{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{21}$$

we aim at sampling $\boldsymbol{u}_{\text{HR}}$ conditioned on $\boldsymbol{u}_{\text{LR}}^{\text{noisy}}$. We consider the equation:

$$\boldsymbol{u}_{\text{LR}}^{\text{noisy}} = \boldsymbol{B} \begin{pmatrix} \boldsymbol{u}_{\text{HR}} \\ \mathbf{n} \end{pmatrix} \tag{22}$$

where $\boldsymbol{B} = \begin{pmatrix} \boldsymbol{A} & \sigma \boldsymbol{I} \end{pmatrix}$ and apply a kriging reasoning on $(\boldsymbol{u}_{\text{HR}}, \mathbf{n}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Gamma} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix}$. A kriging matrix $\boldsymbol{\Delta} = \begin{pmatrix} \boldsymbol{\Delta}_1 & \boldsymbol{\Delta}_2 \end{pmatrix}$ where $\boldsymbol{\Delta}_1 \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M/r,N/r}}$, $\boldsymbol{\Delta}_2 \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M/r,N/r}}$ is solution of the kriging equation

$$\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T \boldsymbol{\Delta} = \boldsymbol{B}\boldsymbol{\Sigma}. \tag{23}$$

It leads to

$$\boldsymbol{\Delta}_1 = (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{A}\boldsymbol{\Gamma}. \tag{24}$$

Note that $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I}$ is invertible because $\sigma > 0$ and $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T$ is a positive matrix. Finally, by denoting $\boldsymbol{\Lambda}^{\text{noisy}} := \Delta_1 = (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{A}\boldsymbol{\Gamma}$, we obtain SR samples as

$$\boldsymbol{u}_{\text{SR}} = \left[\boldsymbol{\Lambda}^{\text{noisy}}\right]^T \boldsymbol{u}_{\text{LR}}^{\text{noisy}} + \tilde{\boldsymbol{u}} - \left[\boldsymbol{\Lambda}^{\text{noisy}}\right]^T (\boldsymbol{A}\tilde{\boldsymbol{u}} + \sigma \tilde{\mathbf{n}}), \quad \tilde{\boldsymbol{u}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}), \tilde{\mathbf{n}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{25}$$

In this noisy case, computing the kriging component is equivalent to first denoise the image with a linear Wiener filter and then apply a noiseless kriging matrix because:

$$\begin{aligned}
\boldsymbol{\Lambda}^T \,\textbf{Wiener}(\boldsymbol{u}_{\text{LR}}^{\text{noisy}}) &= \boldsymbol{\Lambda}^T \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{u}_{\text{LR}}^{\text{noisy}} && (26) \\
&= \boldsymbol{\Gamma}\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T)^\dagger \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{u}_{\text{LR}}^{\text{noisy}} && (27) \\
&= \boldsymbol{C}_t \boldsymbol{M} \left(\boldsymbol{M}^T \boldsymbol{M}\right)^\dagger \boldsymbol{M}^T \boldsymbol{M} (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{u}_{\text{LR}}^{\text{noisy}} \text{ with } \boldsymbol{M} = (\boldsymbol{A}\boldsymbol{C}_t)^T && (28) \\
&= \boldsymbol{C}_t \boldsymbol{M} \boldsymbol{M}^\dagger \boldsymbol{M} (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{u}_{\text{LR}}^{\text{noisy}} \text{ because } \boldsymbol{M}^\dagger = \left(\boldsymbol{M}^T \boldsymbol{M}\right)^\dagger \boldsymbol{M}^T && (29) \\
&= \boldsymbol{C}_t \boldsymbol{M} (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{u}_{\text{LR}}^{\text{noisy}} \text{ because } \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{M} = \boldsymbol{M} && (30) \\
&= \boldsymbol{\Gamma}\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{u}_{\text{LR}}^{\text{noisy}} && (31) \\
&= \left[\boldsymbol{\Lambda}^{\text{noisy}}\right]^T \boldsymbol{u}_{\text{LR}}^{\text{noisy}}. && (32)
\end{aligned}$$

F Proof of Proposition 4

Let $\boldsymbol{\Lambda}^T \in \mathbb{R}^{3\Omega_{M,N} \times 3\Omega_{M/r,N/r}}$ being in the form $\boldsymbol{C}_{\boldsymbol{\eta}}\boldsymbol{S}^T$ where the kernel $\boldsymbol{\eta} \in \mathbb{R}^{3\Omega_{M,N}}$. The law of $\tilde{\boldsymbol{X}} - \boldsymbol{\Lambda}^T(\boldsymbol{A}\tilde{\boldsymbol{X}})$ is invariant by translation by $(rk, r\ell)$ for $k, \ell \in \mathbb{Z}$ since $\tilde{\boldsymbol{X}}$ follows a stationary law, $\boldsymbol{\Lambda}^T\boldsymbol{A}\tilde{\boldsymbol{X}} = \boldsymbol{C}_{\boldsymbol{\eta}} \star \boldsymbol{S}^T(\boldsymbol{A}\tilde{\boldsymbol{X}})$ with $\boldsymbol{A}\tilde{\boldsymbol{X}} \in \mathbb{R}^{3\Omega_{M/r,N/r}}$ following a stationary law.

G Samples of SR algorithms

To illustrate the variety of samples of the stochastic SR algorithm, we show several outputs in the following. Images of Figure 15 are outputs of the stochastic algorithms Gaussian SR (ours), SRFlow, DDRM and MCG with the same HR and reference images than Figures 6 and 7. Similarly, Figure 16 (resp. Figure 17) is related to Figures 8 and 9 with $\sigma = 2/255$ (resp. $\sigma = 10/255$). In this noisy case, samples of the stochastic DDRM, Gaussian SR (ours), BM3D+Gaussian SR (BM3D+GSR) and DPS are presented.

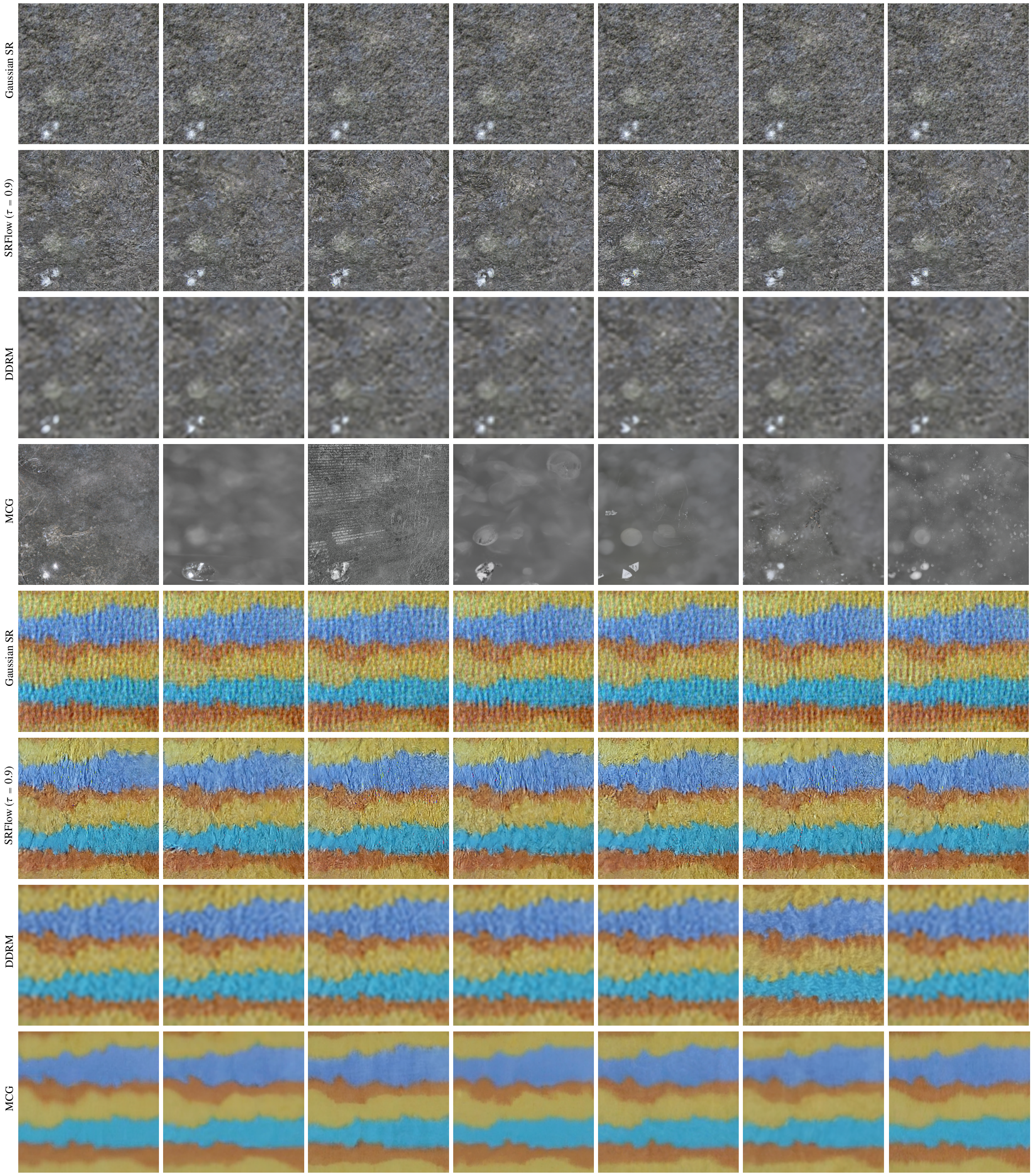


Figure 15: SR methods samples in the context of Figures 6 and 7.

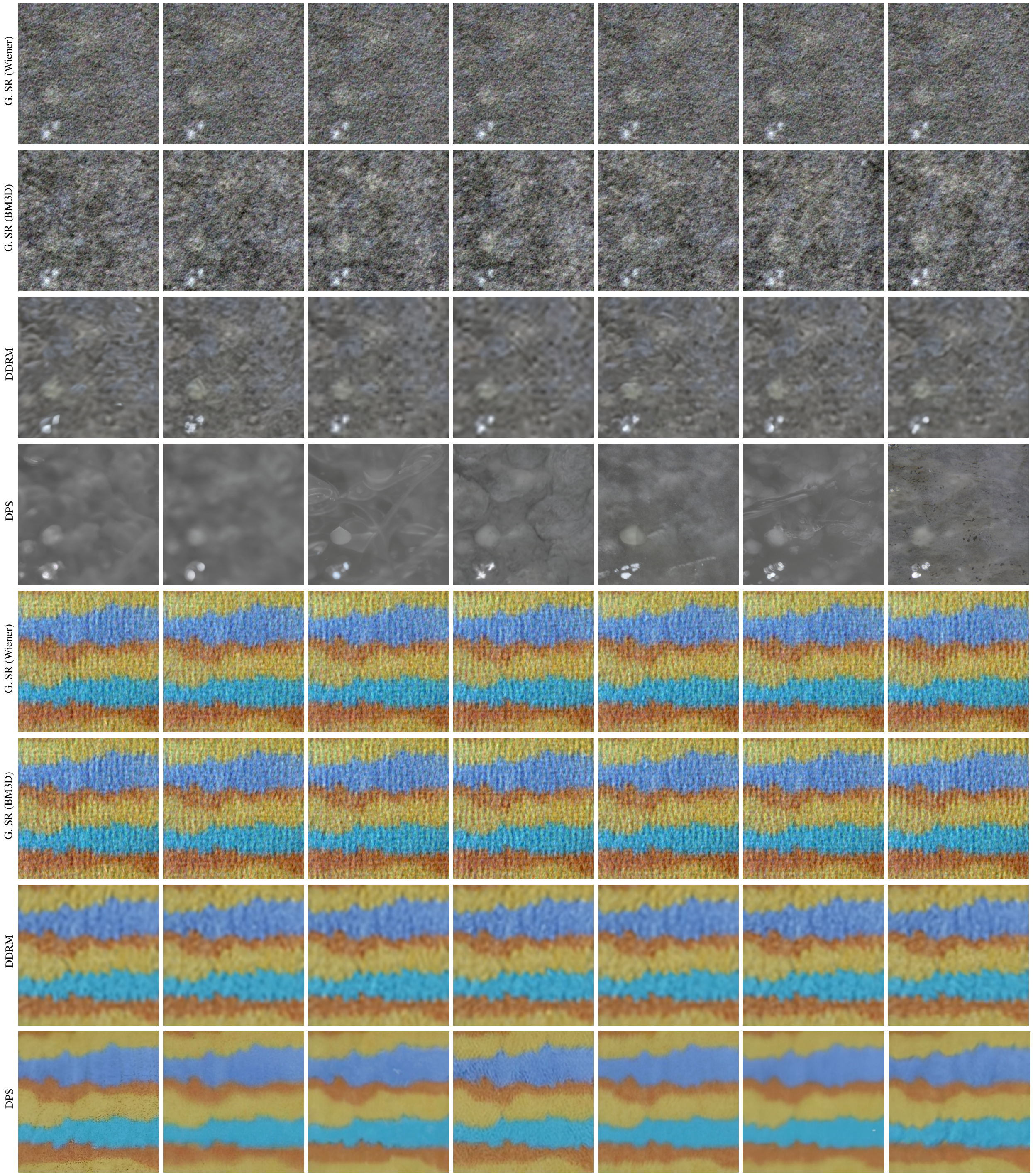


Figure 16: SR methods samples in the context of Figure 8 in a presence of low level noise ($\sigma = 2/255$).

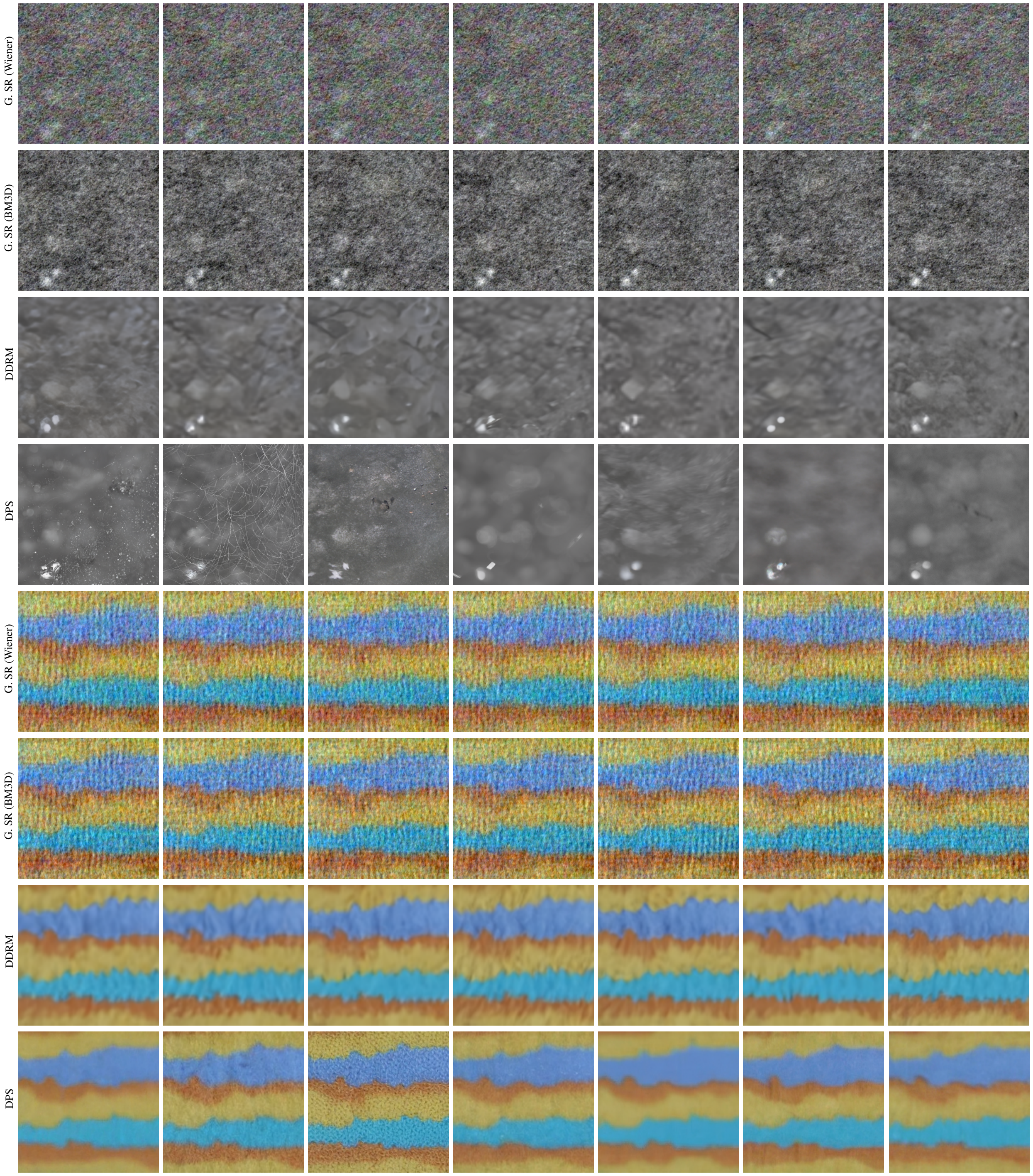


Figure 17: SR methods samples in the context of Figure 9 in a presence of low level noise ($\sigma = 10/255$).