Stochastic super-resolution for Gaussian microtextures
Émile Pierret, Bruno Galerne

To cite this version:
Émile Pierret, Bruno Galerne. Stochastic super-resolution for Gaussian microtextures. 2024. hal-04560456v1

HAL Id: hal-04560456
https://hal.science/hal-04560456v1
Preprint submitted on 26 Apr 2024 (v1), last revised 29 May 2024 (v2)

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

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Stochastic super-resolution for Gaussian textures *

Émile Pierret † Bruno Galerne ‡

April 26, 2024

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.

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 [11, 40, 35, 37]. However, the CNN outputs tend to be blurry as they correspond to the mean of plausible SR solutions [56]. To avoid this regression to the mean issue, other approaches rely on a perceptual loss [4, 59, 52], that is, an $L_2$ loss between pre-trained VGG features [53] introduced in [31], or Generative Adversarial Network (GAN) [23] conditioned on the LR input image [39, 60]. More generally SR is a linear inverse problem that can be tackled by most general deep learning techniques for inverse problems in imaging [47] that go beyond end-to-end learning, notably variational frameworks such as deep image prior [57, 58] and Plug-and-Play...
methods [32]. We do not explore this important body of literature in this work since we are more interested in stochastic methods.

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. [42] use a conditional normalizing flows [36] (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 [55, 26]. The diffusion framework can be used to train conditional diffusion models specific to SR [51]. Alternatively generic diffusion models can be used to provide approximate conditional sampling procedure for inverse problems such as SR [55, 44, 7, 51, 54], an approach that can be extended to latent diffusion models [50]. Alternatively, Jiang et al. [30] 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 [6].

Another approach is to apply reference-based SR [14, 13, 65, 5, 41]. 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 [25]. This variational approach, coined Wasserstein Patch Prior (WPP), exploits important statistics proved to be relevant for texture synthesis [24, 18, 28, 29]. 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 [44, 45]. The WPP framework provides adaptative statistical guarantees to the proven efficiency of patch-based approaches [66, 63], which is critical for the SR of texture images.

In this work, we will solely focus on a precise class of textures, the so-called Gaussian microtextures [19]. 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 [9, 16], can be generated on arbitrary large continuous domains [15, 17], extended to video and visually mixed using Wasserstein barycenters between Gaussian distributions [62]. Closely related to our work, the Gaussian modeling allows for solving texture inpainting via Gaussian conditional sampling based on kriging formulas [22, 21]. This work provides an iterative algorithm based on conjugate gradient descent 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 [49].

1.2 Contributions

Our approach builds on the kriging framework used for Gaussian inpainting [22, 21]. 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 channels, 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 [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 [3] we propose an approximate algorithm for stochastic SR of RGB micro textures 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 [4] where we also compare our results with state-of-the-art methods. This
comparaison leads to a discussion on adapted metrics to evaluate textures SR. Finally, we document the shortcomings of our approach and show that it extends to more general linear operators in [5] and conclude our paper in [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 \( [k] = \{0, \ldots, k - 1\} \) for any integer \( k \geq 1 \), \( \Omega_{M,N} = [M] \times [N] \) denotes the pixel grid, and all the images are extended on \( \mathbb{Z}^2 \) by periodization. For \( u, v \in \mathbb{R}^{\Omega_{M,N}} \), \( u \star v \) designates the discrete and periodic convolution defined by 
\[
(u \star v)(x) = \sum_{y \in \Omega_{M,N}} u(x - y)v(y)
\]
for \( x \in \Omega_{M,N} \). Given a kernel \( t \in \mathbb{R}^{\Omega_{M,N}} \), let us denote \( C_t \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M,N}} \) the matrix associated with the convolution by \( t \) such that for \( u \in \mathbb{R}^{\Omega_{M,N}} \), 
\[
C_t u = t \ast u \in \mathbb{R}^{\Omega_{M,N}}.
\]

The most common zoom-out operator used in the SR literature is the Matlab bicubic zoom-out operator provided by the function \emph{imresize} [39, 60, 3]. Let us denote \( A \) this bicubic zoom-out operator by a factor \( r \). It can be described as a convolution followed by a subsampling \( S \) with stride \( r \) such that for \( u \in \mathbb{R}^{\Omega_{M,N}} \), \( Su \in \mathbb{R}^{\Omega_{M/r,N/r}} \) and for \( x \in \Omega_{M/r,N/r} \), 
\[
(Su)(x) = u(rx) \text{ and for } x \in \Omega_{M/r,N/r}, v \in \mathbb{R}^{\Omega_{M/r,N/r}}, (S^T v)(rx) = v(x), \text{ see e.g. [3].}
\]

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 \emph{imresize} [39, 60, 3]. Let us denote \( A \) this bicubic zoom-out operator by a factor \( r \). It can be described as a convolution followed by a subsampling \( S \) with stride \( r \) such that for \( u \in \mathbb{R}^{\Omega_{M,N}} \), \( Su \in \mathbb{R}^{\Omega_{M/r,N/r}} \) and for \( x \in \Omega_{M/r,N/r} \), 
\[
(Su)(x) = u(rx) \text{ and for } x \in \Omega_{M/r,N/r}, v \in \mathbb{R}^{\Omega_{M/r,N/r}}, (S^T v)(rx) = v(x), \text{ see e.g. [3].}
\]

2.2 The ADSN model associated with a grayscale image

Given a grayscale image \( u \in \mathbb{R}^{\Omega_{M,N}} \) with mean grayscale \( m \in \mathbb{R} \), one defines the Asymptotic Discrete Spot Noise (ADSN) associated with \( u \), \( \text{ADSN}(u) \) as the distribution of \( m1_{\Omega_{M,N}} + t \ast w \) where \( t = \frac{1}{\sqrt{MN}}(u - m1_{\Omega_{M,N}}) \) is called the texton associated with \( u \) and \( w \sim \mathcal{N}(0, I_{\Omega_{M,N}}) \) [19]. Let us recall that in practice it is preferable to replace \( u \) by its periodic component [45] to avoid boundary issues [19]. ADSN \( u \) is a Gaussian distribution with mean \( m1_{\Omega_{M,N}} \) and covariance matrix \( \Gamma = C_tC_t^T = C_t1_{\Omega_{M,N}} \). Samples can be generated fastly executing the convolution with the Fast Fourier Transform (FFT). ADSN \( u \) is stationary, that is, for \( X \sim \text{ADSN}(u) \) and any \( y \in \mathbb{Z}^2 \), \( X(\cdot - y) \) has the same distribution as \( X \). 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 \( t \ast w \sim \mathcal{N}(0, \Gamma) \) that will be called ADSN \( u \) with a slight abuse of notation.
We show in this section that in our SR context, the kriging equation can be solved exactly and fastly in the Fourier domain.

Due to the stationarity of the Gaussian law, Equation 2 needs to be solved by an iterative method, even with the stationarity assumption of the Gaussian law [21].

As presented in 1, the kriging component is a HR blurred version of the LR image with added covariance information.

Let \( X \in \mathbb{R}^{Q_{LR}} \) be a random vector following a Gaussian multivariate law \( \mathcal{N}(0, \Gamma) \) with \( \Gamma \in \mathbb{R}^{Q_{HR} \times Q_{LR}} \) and let \( A \in \mathbb{R}^{Q_{LR} \times Q_{LR}} \) be a linear operator. The simulation of \( X \) knowing \( AX \) can be computed using the following theorem, recalled in [21], which expresses the link between orthogonality and independence for Gaussian vectors.

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

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

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

\[
\Lambda \Gamma A^T = A \Gamma.
\]

2.3 The kriging reasoning

Let \( X \in \mathbb{R}^{Q_{LR}} \) be a random vector following a Gaussian multivariate law \( \mathcal{N}(0, \Gamma) \) with \( \Gamma \) and \( A \) be a linear operator. The simulation of \( X \) knowing \( AX \) can be computed using the following theorem, recalled in [21], which expresses the link between orthogonality and independence for Gaussian vectors.

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

\[
\Lambda \Gamma A^T = A \Gamma.
\]

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 \( \Lambda = (A \Gamma A^T)^\dagger \Lambda \Gamma \) where \((A \Gamma A^T)^\dagger \) is the pseudo-inverse of \( A \Gamma A^T \). The main practical issue is now to compute efficiently \( \Lambda^T v \) for a given LR image \( v \in \mathbb{R}^{Q_{LR}} \). As shown by the following proposition, this simply corresponds to applying a specific convolution.
Moreover, the pseudo-inverse of a convolution is also a convolution. Let \( \Lambda = SC_\epsilon \) and let \( \Gamma = C_\epsilon C_\epsilon^T = C_{\epsilon \epsilon}^\top \) be the covariance of a Gaussian distribution ADSN(u) with associated texton \( t = \frac{1}{\sqrt{M_N}}(u - m\Omega_{MN}) \). \( \Lambda = (\Lambda^\Gamma^T)^\top \Lambda^\Gamma \) is an exact solution of Equation 2 and for all \( v \in \mathbb{R}^{\Omega_{MN}/r} \),
\[
\Lambda^T v = \Lambda \ast (S^T v) \tag{4}
\]
where \( \Lambda = t \ast \tilde{t} \ast \tilde{c} \ast (S^T \kappa^\top) \) with \( \kappa = S(t \ast \tilde{t} \ast \tilde{c} \ast \tilde{c}) \in \mathbb{R}^{\Omega_{MN}/r} \) and \( \kappa^\top \) the convolution kernel defined in Fourier domain by
\[
\hat{\kappa}^\top(\omega) = \begin{cases} \frac{1}{\hat{k}(\omega)} & \text{if } \hat{k}(\omega) \neq 0, \\ 0 & \text{otherwise,} \end{cases} \quad \omega \in \mathbb{R}^{\Omega_{MN}/r}.
\]

**Proof.** Equation 2 is a normal equation associated with the least squares problem \( \text{argmin}_{\Lambda \in \mathbb{R}^{\Omega_{MN}/r} \times \mathbb{R}^{\Omega_{MN}/r}} \| C_\epsilon^T \Lambda - C_{\epsilon \epsilon}^T \|^2_2 \). Consequently, \( \Lambda = (\Lambda^\Gamma^T)^\top \Lambda^\Gamma \) is one of the solutions of Equation 2. Then, we use the following elementary lemma (proved in [3] for completeness).

**Lemma 1** (Convolution and subsampling). 1. If \( C_\alpha \) is the convolution on \( \Omega_{MN} \) by the kernel \( \alpha \in \Omega_{MN} \), \( SC_\alpha S^T \) is the convolution on \( \Omega_{MN/r} \) by the kernel \( S_\alpha \), that is, \( SC_\alpha S^T = C_{S_\alpha} \).

2. If \( C_\beta \) is a convolution on \( \Omega_{MN/r} \) by the kernel \( \beta \in \Omega_{MN/r} \), \( S^T C_\beta = C_{S^T \beta} \).

By Lemma 1, \( \Lambda^\Gamma A^T = S(C_\epsilon \Gamma C_\epsilon^T) S^T \) is a convolution matrix associated with the kernel \( \kappa = S(t \ast \tilde{t} \ast \tilde{c} \ast \tilde{c}) \in \mathbb{R}^{\Omega_{MN}/r} \). Moreover, the pseudo-inverse of a convolution is also a convolution. \( (\Lambda^\Gamma A^T)^\top \) is a convolution associated with the kernel \( \kappa^\top \) such that for \( \omega \in \mathbb{R}^{\Omega_{MN}/r} \), \( \hat{\kappa}^\top(\omega) = \begin{cases} \frac{1}{\hat{k}(\omega)} & \text{if } \hat{k}(\omega) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \)

Also by Lemma 1, \( S^T (\Lambda^\Gamma A^T) = S^T C_{\kappa^\top} = C_{S^T \kappa^\top} \).

\[
\Lambda^T = \Gamma^T A^T (\Lambda^\Gamma A^T)^\top = \Gamma^T C_\epsilon^T S^T C_{\kappa^\top} = \Gamma^T C_\epsilon^T C_{S^T \kappa^\top} S^T = C_\Lambda S^T
\]
where \( \Lambda = t \ast \tilde{t} \ast \tilde{c} \ast \tilde{c} \ast \tilde{c} \ast \tilde{c} \ast \tilde{c} \).

In what follows we refer to \( \Lambda \) as the *kriging kernel.* 2 shows the DFT of the kernel \( \Lambda \) associated with the images of [1]. In contrast with a non-adaptive pseudo-inverse of the bicubic kernel that would be isotropic, we can observe that the kernel \( \Lambda \) is adapted to the covariance structure of the texture and that it amplifies its characteristic frequencies.

To compute SR samples \( u_{SR} \) given by Equation 3, we need to compute \( \Lambda \in \mathbb{R}^{\Omega_{MN}} \) the kernel associated with \( \Lambda^T \) as described in Proposition 1 and then apply \( \Lambda^T \) as described in Equation 2. 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 \( \Lambda^T \) implies the division by \( \hat{\kappa} \) in the Fourier domain. However, in practice, this division can be unstable since a non-zero low value of \( \hat{\kappa} \) amplifies its corresponding frequency. Actually, in our context, \( \Lambda^T \) is applied to LR versions of realizations of an ADSN model that are \( v = A(t \ast w) \) with \( w \in \mathbb{R}^{\Omega_{MN}} \) an unknown white Gaussian noise. Proposition 2 ensures that the application of \( \Lambda^T \) is stable when applied to images that comply with this assumption.

**Proposition 2** (Stability of the kriging operator on the subspace of the LR ADSN samples). Let \( \Lambda^T = \Gamma A^T (\Lambda^\Gamma A^T)^\top \in \mathbb{R}^{\Omega_{MN} \times \Omega_{MN/r}} \). Then,
\[
\forall w \in \mathbb{R}^{\Omega_{MN}}, \| \Lambda^T A(t \ast w) \|_2 \leq \| C_{\lambda} \|_2 \| w \|_2 \leq \| t \|_1 \| w \|_2. \tag{5}
\]
We denote by $3.1$ Description of the framework and notation

### 3 The conditional super-resolution of RGB Gaussian microtextures

**Proof.** As a reminder [48], let $M$ being a real-valued matrix,

$$M^\dagger = \left(M^T M\right)^{-\frac{1}{2}} M^T$$

With $M = AC_i^T$,

$$AC_i^T = (AC_i(AC_i)^T)^{\dagger} AC_i = (A \Gamma A^T)^{\dagger} AC_i$$

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

$$\Lambda^T A(t * w) = \Lambda A^T (\Lambda \Gamma A^T)^{\dagger} AC_i w = C_i(AC_i)^T ((AC_i)^T)^{\dagger} w$$

and

$$\| \Lambda^T A(t * w) \|_2 \leq \|C_i\|_2 \left\| \left((AC_i)^T \left((AC_i)^T\right)^{\dagger} \right. \|w\|_2 \leq \|t\|_1 \|w\|_2, $$

using that for any matrix $M$ one has $\|MM^\dagger\|_2 \leq 1$. $\square$

### 3 The conditional super-resolution of RGB Gaussian microtextures

#### 3.1 Description of the framework and notation

We denote by $3\Omega_{\Omega M}$ the space of the RGB images and by $u_1, u_2, u_3$ the three channels of a given RGB image $u \in \mathbb{R}^\Omega_{\Omega M}$. $	ilde{u}$ designates the Discrete Fourier transform of $u$ defined as the 2D DFT of each channel. The associated texton of an RGB image $u$ is $t = \frac{1}{\sqrt{\Omega M}}(u - m) \in \mathbb{R}^\Omega_{\Omega M}$ where $m \in \mathbb{R}^\Omega_{\Omega M}$ is the mean RGB color. The ADSN model ADSN($u$) is defined as the distribution of $m + \left(t_1 * w \atop t_2 * w \atop t_3 * w\right)$ where $w \sim \mathcal{N}(0, I_{\Omega M})$. Note that the white Gaussian noise is the same in each channel [19]. ADSN($u$) is a multivariate Gaussian law associated with the covariance matrix

$$\Gamma = \begin{pmatrix} C_{t_1} & C_{t_2} & C_{t_3} \\ C_{t_2} & C_{t_3} & C_{t_1} \\ C_{t_3} & C_{t_1} & C_{t_2} \end{pmatrix} \in \mathbb{R}^{\Omega M \times \Omega M}$$

which is the matrix of a multi-channel convolution, that is, each output channel is a linear combination of 2D convolutions of the input RGB channels. More precisely, for all $u \in \mathbb{R}^\Omega_{\Omega M}$,

$$(\Gamma u)_i = t_i * \tilde{t}_1 * u_1 + t_i * \tilde{t}_2 * u_2 + t_i * \tilde{t}_3 * u_3, \quad 1 \leq i \leq 3. \quad (6)$$

Unfortunately, this matrix is not diagonal in the Fourier basis which makes a major difference with the grayscale framework. We still denote by $A = SC_c$ the zoom-out operator that acts on each channel 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 and Lemma are still valid, except that now \(A\Gamma A^T\) and \((A\Gamma A^T)^\dagger\) are multi-channel 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 \((A\Gamma A^T)^\dagger\) turns out to be a critical task. Even for small images, a direct computation of the pseudo-inverse of \(A\Gamma A^T\) with standard routines leads to instabilities. A more principled approach is to remark that for each \(v \in \mathbb{R}^{3M_i/N_i}\) and each \(\omega \in \Omega_{M_i/N_i}\), there exists a \(3 \times 3\) matrix \(\hat{K}(\omega)\) such that \(\mathcal{F}_{\omega}(A\Gamma A^Tv)(\omega) = \hat{K}(\omega)\hat{v}(\omega)\), reducing the problem to computing one \(3 \times 3\) 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 \(X \in \mathbb{R}^{3M_i/N_i}\), following the Gaussian law \(\mathcal{N}(0, \Gamma)\),

\[
\mathbb{E}[X_i | AX_1, AX_2, AX_3] \approx \mathbb{E}[X_i | AX_i], \quad 1 \leq i \leq 3.
\]

The rational behind this approximation is that to reconstruct the channel \(X_i\) from the three channels \(AX = (AX_1AX_2AX_3)\), the more relevant information is in \(AX_i\). 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

\[
\Lambda_{\text{approx}} = \begin{pmatrix}
\Lambda_1 & 0 & 0 \\
0 & \Lambda_2 & 0 \\
0 & 0 & \Lambda_3
\end{pmatrix} \in \mathbb{R}^{3M_i/N_i \times 3M_i/N_i}
\]

as an approximate solution of Equation in the RGB setting, where for each \(1 \leq i \leq 3\), \(\Lambda_i \in \mathbb{R}^{\Omega_{M_i/N_i} \times \Omega_{M_i/N_i}}\) is a solution of

\[
A\Gamma_i A^T \Lambda_i = A\Gamma_i, \quad 1 \leq i \leq 3,
\]

\(\Gamma_i\) being the covariance matrix of the \(i\)th channel. While \(\Lambda_{\text{approx}}\) is not a solution of the kriging equation, it is the kriging operator associated with an ADSN law for which the RGB channels are uncorrelated (which is not an interesting model to generate textures). However, we apply this kriging operator \(\Lambda_i^T\) to textures having properly correlated channels according to the ADSN model \(\mathcal{N}(0, \Gamma)\). More precisely, a RGB Gaussian SR sample \(u_{\text{SR}}\) is defined as

\[
u_{\text{SR},i} = \Lambda_i^T(u_{\text{LR},i} - A\tilde{u}_i) + \tilde{u}_i, \quad 1 \leq i \leq 3,
\]

where, for \(1 \leq i \leq 3\), \(\tilde{u}_i = t_i \ast w\) with \(w \sim \mathcal{N}(0, I_{\Omega_{M_i/N_i}})\) the common noise used for each channel and the multiplication by \(\Lambda_i^T\) 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 [21], 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 \((A\Gamma A^T)^\dagger\varphi\) for a given \(\varphi \in \mathbb{R}^{\Omega_{M_i/N_i}}\). For completeness, the algorithm is recalled in C. This is an iterative algorithm with a stopping criteria \(\epsilon\) and a given number of iterations. The output \(\psi\) of CGD tends to minimize

\[
\|(A\Gamma A^T)^2\varphi - (A\Gamma A^T)\psi\|_2
\]

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 \(10^8\) for fast results and \(10^9\) for high-quality results [20] 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 [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.

In [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 grayscale
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 $512 \times 768$, 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.

Table 1: Quantitative evaluation of the results of 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 [20] for inpainting. The grayscale example confirms 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 [1] producing such an accurate RGB sample with the CGD algorithm is four to five orders of magnitude slower.

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 $\mathbf{u}_{HR}$ has to be known (via the computation of the associated texton $t$). In the experiments of the previous section, the texton is computed from the ground truth HR image $\mathbf{u}_{HR}$, making the algorithm impractical.

From now on, following recent contributions on texture super-resolution [23, 2], we consider that the input image
\( \mathbf{u}_{\text{LR}} \) is given with a companion reference HR image \( \mathbf{u}_{\text{ref}} \) such that both HR images \( \mathbf{u}_{\text{ref}} \) and the unknown \( \mathbf{u}_{\text{HR}} \) are supposed to be realizations of the same texture model. More precisely, in our context based on the ADSN model, Algorithm 1 will be called with input an LR image \( \mathbf{u}_{\text{LR}} \) to be upsampled and the texton \( \mathbf{t}_{\text{ref}} = \frac{1}{\mathbf{M}_{\text{MN}} (\mathbf{u}_{\text{ref}} - m_{\mathbf{1}_{\Omega_{\mathbf{MN}}}}) } \) associated with the reference image \( \mathbf{u}_{\text{ref}} \), meaning that \( \mathbf{u}_{\text{HR}} \) is supposed to be a realization of the ADSN(\( \mathbf{u}_{\text{ref}} \)) law. This makes our routine a stochastic super-resolution algorithm with reference image.

Figure 4 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 4 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.

<table>
<thead>
<tr>
<th>LR image</th>
<th>HR image</th>
<th>Reference image</th>
<th>Sample</th>
<th>Kriging</th>
<th>Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: 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.
Quantitative comparison with the state-of-the-art methods results from Table 2: on a broader range of images. Moreover, our method necessitates a HR reference image. Still it is interesting to compare our method on Gaussian microtextures. This limited scope is a main drawback of Gaussian SR that cannot work on more generic images. However, it does not retrieve correctly the details, as it can be observed with the white artefacts generally observed in networks’ outputs. WPP generates textured but too smooth image. We can also observe SRFlow with a positive temperature gives good textured solutions but with frequently do not really recover the texture. 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 for solving the SR problem. A step of attachment to degraded data is added in the generation to solve inverse problems such that it depends on a hyperparameter $\tau$, that modulates the variance of the latent space. DPS and DDRM proposes also a strong attachment to data, illustrated by the high LR-PSNR metric. It is measured 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 SIMilarity(SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). The PSNR is the logarithmic scale of the mean-square error (MSE) between the HR image $u_{HR}$ 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 $u_{LR}$. The SSIM quantifies the similarity between images relying on a luminance, a contrast and a structure terms. LPIPS 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 , DDRM tends to produce blurred samples and DPS smooth images and consequently do not really recover the texture. SRFlow with a positive temperature gives good textured solutions but with artefacts generally observed in networks’ outputs. WPP generates textured but too smooth image. We can also observe that the different luminance of the reference image in 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 .

Table 2: Quantitative comparison with the state-of-the-art methods results from ?? realized on 100 samples for stochastic methods. Blurry results outperforms 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.

### 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 ?? . Let us describe the concurrent algorithms and the metrics used for comparison.

Super-Resolution using Normalizing Flow (SRFlow)\(^1\), Denoising Diffusion Restoration Models (DDRM)\(^2\) and Diffusion Posterior Sampling (DPS)\(^3\) are diverse super-resolution routines where several samples are proposed to solve the inverse problem \(^4\). 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 to solve the SR problem. It depends on a hyperparameter $\tau$, the temperature, that modulates the variance of the latent space. DPS and DDRM work with a diffusion model that has been trained on the dataset ImageNet\(^5\) for image generation during several days on GPU\(^6\).

<table>
<thead>
<tr>
<th>Model</th>
<th>PSNR</th>
<th>LR-PSNR</th>
<th>SSIM</th>
<th>LPIPS</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian SR</td>
<td>17.05</td>
<td>-0.04</td>
<td>0.15</td>
<td>-0.02</td>
<td>-</td>
</tr>
<tr>
<td>Bicubic</td>
<td>19.74</td>
<td>0.19</td>
<td>0.18</td>
<td>-0.18</td>
<td>-</td>
</tr>
<tr>
<td>WPP</td>
<td>20.14</td>
<td>0.15</td>
<td>0.18</td>
<td>-0.15</td>
<td>-</td>
</tr>
<tr>
<td>SRFlow ($\tau=0.2$)</td>
<td>21.44</td>
<td>0.20</td>
<td>0.20</td>
<td>-0.20</td>
<td>-</td>
</tr>
<tr>
<td>SRFlow ($\tau=0.9$)</td>
<td>21.75</td>
<td>0.20</td>
<td>0.20</td>
<td>-0.20</td>
<td>-</td>
</tr>
<tr>
<td>DDRM</td>
<td>21.57</td>
<td>0.17</td>
<td>0.17</td>
<td>-0.17</td>
<td>-</td>
</tr>
<tr>
<td>DPRS</td>
<td>20.59</td>
<td>0.12</td>
<td>0.12</td>
<td>-0.12</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^1\)Code and weights from https://github.com/andreas128/SRFlow
\(^2\)Code and weights from https://github.com/bahjat-kawar/ddrm
\(^3\)Code and weights from https://github.com/Johertrich/Wasserstein_Patch_Prior

2. Code and weights from https://github.com/bahjat-kawar/ddrm

---

1. Gaussian SR (ours) 17.05 ± 0.04 159.24 ± 0.04 0.08 ± 0.01 0.22 ± 0.01 0.01\(^7\) 19.14 ± 0.09 154.52 ± 0.36 0.20 ± 0.01 0.23 ± 0.01 0.01\(^7\)
2. Kriging comp. 18.76 159.30 0.11 0.75 - 21.42 154.47 0.30 0.52 -
3. WPP 20.14 0.15 0.18 0.18 20.14 0.15 0.18 0.18 - -
4. SRFlow ($\tau=0.2$) 21.44 0.20 0.20 0.20 21.64 0.20 0.20 0.20 - -
5. SRFlow ($\tau=0.9$) 21.75 0.20 0.20 0.20 21.95 0.20 0.20 0.20 - -
6. DDRM 21.57 0.17 0.17 0.17 21.75 0.17 0.17 0.17 - -
7. DPRS 20.59 0.12 0.12 0.12 20.74 0.12 0.12 0.12 - -
Figure 5: Comparison of our method with the stochastic SRflow, DPS and DDRM and with the SISR WPP that uses the reference image (The HR size is $256 \times 256$ 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: Comparison of our methods with stochastic SRFlow, DPS and DDRM and with the SISR WPP that uses the reference image (The HR size is $256 \times 256$ and $r = 8$). Contrary to WPP, Gaussian SR is not influenced by the difference of luminance between the HR and the reference images.
well-known that PSNR promotes blurry images, a limitation called the “regression to the mean” problem \cite{65} in the SR literature. Our image with the best PSNR comparison is the kriging component. We can explain it theoretically by the next Proposition \cite{3} which shows that in expectation the best PSNR is provided by the mean of our samples, the kriging component.

**Proposition 3 (Kriging component and MSE).** Let $u_{HR} \in \mathbb{R}^{\Omega_{HR}}$ be a HR image, $u_{LR} = Au_{HR}$ its LR version, $A \in \mathbb{R}^{\Omega_{LR} \times \Omega_{HR}}$ be the kriging operator and $X_{SR}$ the random image following the distribution of the SR samples generated with Equation \cite{3} then

$$E_{X_{SR}} (\|u_{HR} - X_{SR}\|^2) = \|u_{HR} - A^T u_{LR}\|^2 + \text{Tr} \left( (I_{\Omega_{HR}} - A^TA)(I_{\Omega_{LR}} - A^TA)^T \right)$$

$$\geq \|u_{HR} - A^T u_{LR}\|^2.$$

Simply put, the expected mean square error between the optimal HR image and Gaussian SR samples is always higher than the MSE between $u_{HR}$ and the associated kriging component $A^T u_{LR}$.

Proposition \cite{3} proved in \cite{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 \cite{27}. 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.

## 5 Limitations and extensions

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 \cite{7}. 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 \cite{7}. The 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 \cite{19}.

### 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($u_{ref}$). \cite{8} 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 \cite{2} does not hold for the simulation of the kriging component. Algorithm \cite{1} can lead to instabilities as illustrated in \cite{9}. 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 \cite{2}. This type of observations is marginal but can occur in some rare cases.
Figure 7: 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 8: 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.

Figure 9: 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.
Figure 10: Variance maps of the different algorithms for stochastic super-resolution applied to the HR image of 5 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 channels is displayed. While DDRM is close to being deterministic, the SRFlow and DPS algorithms have an adaptive variance to the HR image, showing the white spot of the image. In comparison, the Gaussian SR has a cyclostationary non-adaptative variance.

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 $\mathbf{A}_{\text{approx}}^T$ is in the form $\mathbf{G} \mathbf{A}^T C_\eta$ where $C_\eta$ is a diagonal multi-channel convolution but this is also the case with the perfect solution of Equation $2 \mathbf{A} = \mathbf{G} \mathbf{A}^T (\mathbf{A} \mathbf{G} \mathbf{A}^T)^T$. This proposition shows that the images sampled with Gaussian SR have a constant variance on sub-grids of the image domain. The law of the SR samples $u_{\text{SR}}$ inherits the stationarity of ADSN($u_{\text{ref}}$) being invariant by the translations by $(k, \ell)$ where $k, \ell \in \mathbb{Z}$. In other words, the law of our SR samples is a cyclostationary law [43].

Proposition 4 (Non-adaptative variance of Gaussian SR). Let $\mathbf{A}^T \in \mathbb{R}^{3M \times 3N}$ being in the form $C_\eta \mathbf{S}$ where the kernels $\eta \in \mathbb{R}^{3M \times 3N}$. Then, the law of the SR samples generated by Equation 3 is invariant by translations by $(k, \ell)$ where $k, \ell \in \mathbb{Z}$.

Proposition 4 implies that the samples generated by Gaussian SR have a pixelwise variance constant by translations $(k, \ell)$ 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 10. The white spot detail in the textures is taking into 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. DPS variance map show the white spot detail but one can guess that the texture is not really retrieved because the variance is smooth on the rest of the image. DDRM is quasi deterministic with a very low variance value. 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.

5.4 Extension to other degradation operators

Our method can be applied to reconstruct images degraded by other degradation operators in the form $\mathbf{S} \mathbf{C}$ where $\mathbf{C}$ is a convolution preserving the mean. All the proofs and the properties still hold. To illustrate it, we consider motion blurs randomly generated by the code 5 with kernel size $61 \times 61$ and intensity value 0.5, following [7]. We propose to reconstruct images degraded by a motion blur followed by a subsampling operator $\mathbf{S}$ with stride $r = 4$ and images degraded by a motion blur and the bicubic zoom-out operator $\mathbf{A}$ with factor $r = 4$. These are not standard problems but they can be seen as multiple problems as studied in [54]. In 11 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.

https://github.com/LeviBorodenko/motionblur
Figure 11: 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 $r = 4$. 
6 Conclusion

This work provides an efficient sampling algorithm for solutions of the stochastic SR problem for Gaussian micro-textures. Our solution is exact for grayscale images and 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 micro-textures accompanied with a reference HR image. 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 micro-textures, even in the presence of motion blur.

References


### 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. [12]). Let us denote \((\lambda(x))_{x \in \Omega_{M,N}}\) the columns of \(\Lambda\). \(\Lambda\) minimizes

\[
\mathbb{E} \left( \| A^T A X - X \|^2 \right) = \mathbb{E} \left[ \sum_{x \in \Omega_{M,N}} \left( (A^T A X)(x) - X(x) \right)^2 \right] = \sum_{x \in \Omega_{M,N}} \mathbb{E} \left( (\lambda(x)^T A X - X(x))^2 \right).
\]

This is a separable function of the columns \((\lambda(x))_{x \in \Omega_{M,N}}\). For each \(x \in \Omega_{M,N}\), let us minimize \(\lambda(x) \in \mathbb{R}^{M_x \times \delta_x} \rightarrow \mathbb{E} \left( (\lambda(x)^T A X - X(x))^2 \right)\). One has

\[
\mathbb{E} \left( (\lambda(x)^T A X - X(x))^2 \right) = \text{Var}(\lambda(x)^T A X) + \text{Var}(X(x)) - 2 \text{Cov}(\lambda(x)^T A X, X(x)) = \lambda(x)^T A \Gamma A^T \lambda(x) + \text{Var}(X(x)) - 2 \lambda(x)^T A \text{Cov}(X, X(x)).
\]

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

\[A \Gamma A^T \Lambda = A \Gamma.\]

### 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 \(\delta_x^\prime\) for \(x \in \Omega_{M/r,N/r}\). Denoting \(T_x\) the translation by \(x \in \Omega_{M,N}\), that is for all \(z \in \mathbb{R}^{\Omega_{M,N}}, T_x u(z) = u(z - x)\). Remark that convolving with \(\delta_x^{\Omega_{M,N}}\) corresponds to translating by \(x\): for all \(u \in \mathbb{R}^{\Omega_{M,N}}, u \ast \delta_x = T_x u\).

1. For \(x \in \Omega_{M/r,N/r}\),

\[
S C_a S^T \delta_x^\prime = \delta_{T_a x} = S(\delta_x \ast \delta_{T_a x}) = S T_{a x} \delta_x = T_{a x} S \delta_x = (S \delta_x) \ast \delta_x^\prime.
\]

2. For \(x \in \Omega_{M/r,N/r}\),

\[
S^T C \beta \delta_x^\prime = S^T (\beta \ast \delta_x^\prime) = S^T T_a \beta = T_{r x} S^T \beta = (S^T \beta) \ast \delta_{r x} = (S^T \beta) \ast (S^T \delta_x^\prime).
\]

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 [33]. Algorithm 2 aims at solving the normal equation

\[B^T B \psi = B^T \varphi\]
associated with the least-squares problem $\psi \mapsto \|B\psi - \varphi\|_2$ and has good convergence properties when applied to the inpainting problem for Gaussian textures [21]. First we observed that this also applies in the case of super-resolution. In our SR context, our goal is to express the inpainting problem for Gaussian textures [21]. Therefore, we apply CGD with $B = \Lambda^T A^T$ and $\varphi = v$.

Algorithm 2 CGD Algorithm CGD to compute $B^T \varphi$

Input: Initialize $k \leftarrow 0$, $\psi_0 \leftarrow 0$, $r_0 \leftarrow B^T \varphi - B^T B \psi_0$, $d_0 \leftarrow r_0$

while $\|r_k\|_2 > \varepsilon$ do

\begin{align*}
\alpha_k & \leftarrow \frac{\|r_k\|_2}{\|r_k\|_2^2} \\
\psi_{k+1} & \leftarrow \psi_k + \alpha_k d_k \\
r_{k+1} & \leftarrow r_k - \alpha_k B^T B d_k \\
d_{k+1} & \leftarrow r_{k+1} + \frac{\|r_{k+1}\|_2}{\|r_k\|_2} d_k \\
k & \leftarrow k + 1
\end{align*}

end while

Output: $\psi_k$

D Proof of Proposition 3

Let $u_{HR} \in \mathbb{R}^{\Omega_{HR}}$ be a HR image, $u_{LR} = A u_{HR}$ its LR version, $u_{ref}$ a reference image, $\Gamma$ such that ADSN($u_{ref}$) = $\mathcal{N}(0, \Gamma)$, $\Lambda \in \mathbb{R}^{\Omega_{LR, u} \times \Omega_{LR, X}}$ be the kriging operator and $X_{SR}$ the random image following the distribution of the SR samples generated with Equation 21 $X_{SR}$ has the same law as:

$$u_{SR} = \Lambda^T u_{LR} + \tilde{X} - \Lambda^T A \tilde{X}$$

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

$$\mathbb{E}(X_{SR} (\|u_{HR} - X_{SR}\|_2^2)) = \mathbb{E}(u_{HR} - \Lambda^T u_{LR})^2 + \mathbb{E}(\|\tilde{X} - \Lambda^T A \tilde{X}\|_2^2)$$

$$+ 2\langle u_{HR} - \Lambda^T u_{LR}, \mathbb{E}(\tilde{X}) - \Lambda^T A \mathbb{E}(\tilde{X}) \rangle$$

$$= \|u_{HR} - \Lambda^T u_{LR}\|_2^2 + \text{Tr} \left[ (I_{\Omega_{\Lambda, X}} - \Lambda^T A) \Gamma (I_{\Omega_{\Lambda, X}} - \Lambda^T A)^T \right]$$

And, $(I_{\Omega_{\Lambda, X}} - \Lambda^T A) \Gamma (I_{\Omega_{\Lambda, X}} - \Lambda^T A)^T$ is a positive semi-definite matrix with non-negative trace which gives

$$\mathbb{E}(X_{SR} (\|u_{HR} - X_{SR}\|_2^2)) \geq \|u_{HR} - \Lambda^T u_{LR}\|_2^2. \quad (16)$$

E Proof of Proposition 4

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