Stochastic Super-Resolution For Gaussian Textures

Émile Pierret, Bruno Galerne

STOCHASTIC SUPER-RESOLUTION FOR GAUSSIAN TEXTURES

Émile Pierret[a], *Bruno Galerne*[a,b]

[a] Institut Denis Poisson – Université d'Orléans, Université de Tours, CNRS
[b] Institut universitaire de France (IUF)

ABSTRACT

Super-resolution (SR) is an ill-posed inverse problem which consists in proposing high-resolution images consistent with a given low-resolution one. While most SR algorithms are deterministic, stochastic SR deals with designing a stochastic sampler generating any realistic SR solution. The goal of this paper is to show that stochastic SR is a well-posed and solvable problem when restricting to Gaussian stationary textures. Using Gaussian conditional sampling and exploiting the stationarity assumption, we propose an efficient algorithm based on fast Fourier transform. We also demonstrate the practical relevance of the approach for SR with a reference image. Although limited to stationary microtextures, our approach compares favorably in terms of speed and visual quality to some state of the art methods designed for a larger class of images.

***Index Terms*—** stochastic super-resolution, Gaussian textures, conditional simulation, kriging, super-resolution with a reference image

1. INTRODUCTION

The super resolution (SR) problem consists in generating a high-resolution (HR) image corresponding to a given low-resolution (LR) input image. This is a very ill-posed inverse problem that necessitates an image prior model or additional information to create images having sharp edges and rich texture details. This is a very important problem for the entertainment industry due to the increase in resolution of display screens as well as other imaging industries where resolution is critical, notably satellite imaging and microscopy.

Different frameworks are considered in the literature: the Single Image SR (SISR) consists in using only an LR input image and a generic database of HR images, while SR with a reference image considers an LR input image accompanied with an HR image that presents similarity with the unknown HR input. Most contributions in SISR are based on conditional generative neural networks [1, 2, 3, 4], adversarial [5] or not, and trained using the "perceptual loss" [6], that is, the distance between pretrained VGG features [7]. These references tackle zoom factor x4 or even x8. Contrary to a classical x2 problem that can be seen as an image sharpener problem, for such high zoom factors new image content must be generated in accordance with the LR input, and the space of possible images becomes very large. All references then insist on the importance of generating local textures and avoiding the "regression to the mean problem": when favoring PSNR the optimal result consists in a blurry image close to the mean of plausible sharp images. While SISR can be seen as local conditional texture synthesis, a close inspection of the state-of-the-art techniques shows however that texture modeling is absent from GANs. Indeed, the perception loss does not favor the statistics of the textures one wishes to reconstruct. In addition, the proposed models are generally deterministic, which is not desirable for texture generation since it does not allow checking the statistical consistency of the proposed solution. More recent models propose generative networks with stochastic response for SISR: SRFlow [8] use invertible generative flow [9] and [10] use Denoising Diffusion Probabilistic Models (DDPM) [11], also called score-based models [12]. They propose practical solutions for stochastic SR, that is providing a stochastic sampler that outputs any realistic solution to the SISR problem. In addition, let us note that several recent contributions solely focus on texture SR, using patch statistics from a reference image [13, 14] or a prior as spectral content [15].

In this paper, we solve the problem of stochastic SR when restricting to stationary Gaussian textures [16]. Following a similar approach as for Gaussian texture inpainting [17, 18], assuming that the LR input is stationary and Gaussian with a known covariance in HR space, we propose an exact SR sampler relying on conditional Gaussian simulation. We exploit the stationarity of both the texture model and the zoom-out operator to obtain an efficient algorithm based on Fast Fourier Transform (FFT). We further demonstrate experimentally the interest of the approach in the context of SR with a reference image.

The plan of the paper is as follows. We present our framework of SR and remind results about conditional Gaussian simulation. Then, we focus on Gaussian stationary textures and detail how to implement efficiently the stochastic simulation for these models. Finally, we extend the approach for SR with a reference image and compare our results with the state of the art before concluding.

2. GAUSSIAN CONDITIONAL SIMULATION FOR SUPER-RESOLUTION

2.1. Framework

Notations Let $M, N > 2$ be the size of the images. We extend all the images on $\mathbb{Z}^2$ by periodization. We write $[M] = \{0, \dots, M-1\}$ and $\Omega_{M,N} = [M] \times [N]$. For $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{\Omega_{M,N}}$, $\mathbf{X} \star \mathbf{Y}$ designates the discrete and periodic convolution defined by $(\mathbf{X} \star \mathbf{Y})(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_{M,N}} \mathbf{X}\big(\mathbf{x}-\mathbf{y}\big)\mathbf{Y}(\mathbf{y})$, $\mathbf{x} \in \Omega_{M,N}$. For $\boldsymbol{\lambda} \in \mathbb{R}^{\Omega_{M,N}}$, we denote the symmetric $\check{\boldsymbol{\lambda}}$ such that for $\mathbf{x} \in \Omega_{M,N}$, $\check{\boldsymbol{\lambda}}(\mathbf{x}) = \boldsymbol{\lambda}(-\mathbf{x})$. For $\mathbf{X} \in \mathbb{R}^{\Omega_{M,N}}$, we express by $\mathscr{F}_2(\mathbf{X})$ or $\widehat{\mathbf{X}}$ its discrete Fourier transform defined by $\widehat{\mathbf{X}}(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_{M,N}} \mathbf{X}(\mathbf{y}) \exp(-\frac{2i\pi x_1 y_1}{M}) \exp(-\frac{2i\pi x_2 y_2}{M})$, $\mathbf{x} \in \Omega_{M,N}$. Let us recall that $\mathscr{F}_2$ is invertible with $\mathscr{F}_2^{-1} = \frac{1}{MN}\overline{\mathscr{F}_2}$ and that $\mathscr{F}_2(\mathbf{X} \star \mathbf{Y}) = \mathscr{F}_2(\mathbf{X}) \odot \mathscr{F}_2(\mathbf{Y})$ where $\odot$ denotes the componentwise product.

Zoom-out operator We denote by r the factor of zoom-out ($= 4, 8, \dots$). From now on we assume that M and N are multiples of r. The loss of pixels in an image can be caused by multiple operators. We will assume in our experiments that the degradation is caused by the imresize function of Matlab which is used in many mod-

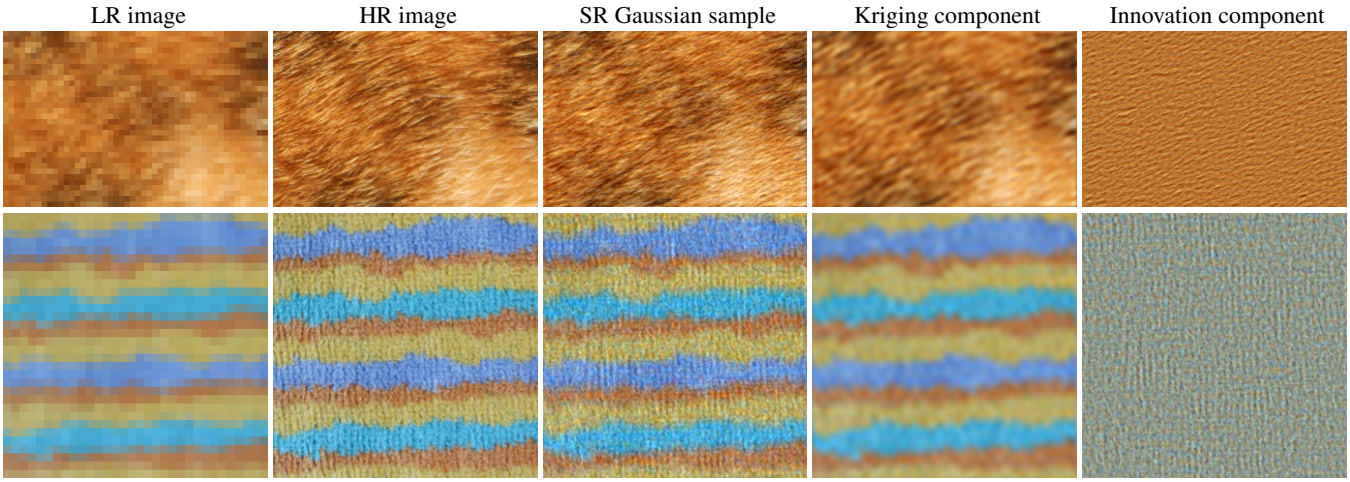


Fig. 1. Different components of the Gaussian SR model. Top row: SR with a factor $\times 16$, the images HR size is 512×768; Bottom row: SR with a factor $\times 8$, the images HR size is 256×256. The last columns presents respectively a Gaussian SR sample and its associated kriging and innovation components. Observe the complementarity of the the kriging component which deblurs the LR input and the innovation component which adds independent high-frequency details.

els [3, 4, 8]. Concretely, this operator reduces the size of an image by weighting the pixels. If we suppose that the image is periodic, we can split it off in a separable convolution which weights all the pixels followed by a subsampling with stride r. We denote $\mathbf{A} = \mathbf{SC}$ this linear zoom-out operator, with $\mathbf{C}$ the convolution and $\mathbf{S}$ the subsampling operator with stride r such that for $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$, $\mathbf{SU} \in \mathbb{R}^{\Omega_{M/r,N/r}}$ and for $\mathbf{x} \in \Omega_{M/r,N/r}$, $(\mathbf{SU})(\mathbf{x}) = \mathbf{U}(r\mathbf{x})$.

2.2. Gaussian conditional simulation

Let $\mathbf{X} \in \mathbb{R}^{\Omega_{M,N}}$ be a stationary Gaussian process with distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{X}_{\mathrm{LR}} = \mathbf{AX} \in \mathbb{R}^{\Omega_{M/r,N/r}}$ be its LR version where $\mathbf{A}$ is the periodic zoom-out operator. $\mathbf{A}$ reduces the size of the images and thus is not invertible. We would like to sample $\mathbf{X}|\mathbf{X}_{\mathrm{LR}}$. In our further assumptions, $\mathbf{X}$ is supposed to be Gaussian, consequently we will use the classical theorem recalled in [18]:

Theorem 1 (Gaussian simulation and Gaussian kriging)**.** *Let $\mathbf{X}$ be a **Gaussian** vector, and $\mathbf{A}$ be a linear operator, $\mathbb{E}(\mathbf{X}|\mathbf{AX})$ and $\mathbf{X} - \mathbb{E}(\mathbf{X}|\mathbf{AX})$ are independent. Consequently, if $\tilde{\mathbf{X}}$ is independent of $\mathbf{X}$ with the same distribution, then $\mathbb{E}(\mathbf{X}|\mathbf{AX}) + (\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}}))$ has the same distribution as $\mathbf{X}$, knowing $\mathbf{AX}$. Furthermore, if $\mathbf{X}$ is zero-mean, there exists $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ such that $\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T\mathbf{AX}$ and $\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T\mathbf{AX}$ if and only if $\mathbf{\Lambda}$ verifies the matrix equation:*

$$\mathbf{A\Gamma A}^T\mathbf{\Lambda} = \mathbf{A\Gamma}. \tag{1}$$

This theorem is a consequence of the fact that the space of square-integrable random variables is a Hilbert space. The assumption that $\mathbf{X}$ follows a Gaussian law is crucial. As for now, the resolution of our problem requires the computation of $\mathbb{E}(\mathbf{X}|\mathbf{X}_{\mathrm{LR}}) = \mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T\mathbf{AX} = \mathbf{\Lambda}^T\mathbf{X}_{\mathrm{LR}}$, where $\mathbf{\Lambda}$ is called the kriging matrix. $\mathbf{\Lambda}^T\mathbf{X}_{\mathrm{LR}}$ is called the kriging component and $\tilde{\mathbf{X}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{X}}$ the innovation component. Therefore, stochastic SR for Gaussian textures corresponds to simulate $\mathbf{\Lambda}^T\mathbf{X}_{\mathrm{LR}} + (\tilde{\mathbf{X}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{X}})$. Figure 1 presents the realization of $\mathbf{X}|\mathbf{X}_{\mathrm{LR}}$, titled SR Gaussian sample. This is the sum of the kriging component which is an attachment to data and the innovation component that adds stochastic independent high-frequency variations. For the given LR image $\mathbf{X}_{\mathrm{LR}}$ and the HR covariance $\mathbf{\Gamma}$, it is just necessary to compute $\mathbf{\Lambda}$ to sample HR versions of $\mathbf{X}_{\mathrm{LR}}$.

3. AN EFFICIENT IMPLEMENTATION IN THE STATIONARY CASE

3.1. ADSN model and super-resolution

Our objective is to study the SR of Gaussian stationary textures. We will focus on textures following the asymptotic discrete spot noise (ADSN) model [16]. Given a grayscale image $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ with mean grayscale $m \in \mathbb{R}$, one defines $\mathrm{ADSN}(\mathbf{U})$ as the distribution of $\mathbf{X} = \mathbf{t} \star \mathbf{W}$ where $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ is called the texton associated with $\mathbf{U}$. $\mathrm{ADSN}(\mathbf{U})$ is a Gaussian distribution with mean zero and covariance matrix $\mathbf{\Gamma}$ that represents the convolution by the kernel $\boldsymbol{\gamma} = \mathbf{t} \star \check{\mathbf{t}}$. $\mathbf{X} \sim \mathrm{ADSN}(\mathbf{U})$ is stationary since for all $\mathbf{x} \in \Omega_{M,N}$ and $\mathbf{y} \in \mathbb{Z}^2$, $\mathbf{X}(\mathbf{x} - \mathbf{y}) \overset{\mathscr{L}}{=} \mathbf{X}(\mathbf{x})$.

For a given input image $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ and its LR version $\mathbf{U}_{\mathrm{LR}} = \mathbf{AU}$, we would like to sample $\mathbf{X} \sim \mathrm{ADSN}(\mathbf{U})$ conditioned on $\mathbf{X}_{\mathrm{LR}} = \mathbf{U}_{\mathrm{LR}}$, that is, by Theorem 1,

$$\mathbf{X}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}} + (\tilde{\mathbf{X}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{X}})$$

where $\tilde{\mathbf{X}} \sim \mathrm{ADSN}(\mathbf{U})$, and $\mathbf{\Lambda}$ is the kriging matrix associated with $\mathrm{ADSN}(\mathbf{U}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$. We will explain how to implement it efficiently in the following.

3.2. Structure of the problem induced by stationarity

For $k, \ell \in [r]$, let $\Omega^{k,\ell,r}_{M,N} = \{(k + ir, \ell + jr),\ i, j \in [M/r] \times [N/r]\} \subset \Omega_{M,N}$ be the subgrid of $\Omega_{M,N}$ having stride r and starting at (k, ℓ). Note that each subgrid $\Omega^{k,\ell,r}_{M,N}$ has the same number of pixels as the LR image domain $\Omega_{M/r,N/r}$.

Proposition 1 (Structure of the kriging matrix)**.** $\mathbf{Y} \in \mathbb{R}^{\Omega_{M/r,N/r}} \mapsto \mathbf{\Lambda}^T\mathbf{Y} \in \mathbb{R}^{\Omega_{M,N}}$ *corresponds to a convolution on each of the shifted subgrids $\Omega^{k,\ell,r}_{M,N}$, $k, \ell \in [r]$. More precisely, $\mathbf{\Lambda}$ is fully determined by its r^2 first columns $\boldsymbol{\lambda}(k, \ell) = \mathbf{\Lambda}_{\Omega_{M/r,N/r} \times (k,\ell)}$, $k, \ell \in [r]$ and*

$$\left(\mathbf{\Lambda}^T\mathbf{Y}\right)(\Omega^{k,\ell,r}_{M,N}) = \check{\boldsymbol{\lambda}}(k, \ell) \star \mathbf{Y}.$$

Input: An image $\mathbf{U}_{\mathrm{LR}} \in \mathbb{R}^{\Omega_{M/r,N/r}}$, r the zoom factor, $\mathbf{t}$ the convolution kernel of the ADSN model, $\mathbf{c}$ the kernel of the convolution of the zoom-out operator $\mathbf{A} = \mathbf{SC}$

Step 1: *Computation of kriging matrix* $\mathbf{\Lambda}$

Store $\mathbf{per}(\boldsymbol{t})$ the periodic component of $\mathbf{t}$;

Store the convolution kernels $\boldsymbol{\gamma} = \mathbf{per}(\boldsymbol{t}) \star \check{\mathbf{per}}(\boldsymbol{t})$, $\mathbf{c} \star \boldsymbol{\gamma}$ and $\boldsymbol{\kappa} = \mathbf{c} \star \boldsymbol{\gamma} \star \check{\mathbf{c}}$ (computed in Fourier);

for $(k,\ell) \in [r]^2$ **do**

$$\widehat{\mathbf{b}} = \mathscr{F}_2\Big(\mathbf{S}((\mathbf{c} \star \boldsymbol{\gamma})(\cdot - k, \cdot - \ell)\Big) ;$$

$$\widehat{\boldsymbol{\lambda}}(k,\ell)\,[\widehat{\boldsymbol{\kappa}} \neq 0] \leftarrow \frac{\widehat{\mathbf{b}}[\widehat{\kappa}\neq 0]}{\widehat{\boldsymbol{\kappa}}[\widehat{\boldsymbol{\kappa}}\neq 0]};$$

end

Step 2: *Sampling of one SR version of* $\mathbf{U}_{\mathrm{LR}}$;

Generate $\mathbf{W} \in \mathbb{R}^{\Omega_{M,N}}$ following a Gaussian standard law;

$\tilde{\mathbf{X}} \leftarrow \mathbf{t} \star \mathbf{W}$;

$\tilde{\mathbf{X}}_{\mathrm{LR}} \leftarrow \mathbf{A}\tilde{\mathbf{X}}$;

for *each shifted subgrid by* $(k,\ell) \in [r]^2$ **do**

$$\mathbf{X}_{\mathrm{SR}}(\Omega^{k,\ell,r}_{M,N}) \leftarrow \mathscr{F}_2^{-1}\left((\widehat{\mathbf{U}_{\mathrm{LR}}} - \widehat{\tilde{\mathbf{X}}}_{\mathrm{LR}}) \odot (\overline{\widehat{\boldsymbol{\lambda}}(k,\ell)})\right) + \tilde{\mathbf{X}}(\Omega^{k,\ell,r}_{M,N});$$

end

Output: $\mathbf{X}_{\mathrm{SR}}$

Algorithm 1: Pseudo-code of the Gaussian simulation for SR. To generate several samples, only the step 2 should be re-run.

The proof of Proposition 1 relies on the fact that both $\mathbf{X}$ and $\mathbf{X}_{\mathrm{LR}} = \mathbf{AX}$ are stationary on their respective domains. Consequently, it is sufficient to solve the r^2 equations

$$\mathbf{A\Gamma A}^T \boldsymbol{\lambda}(k,\ell) = \mathbf{A\Gamma}_{\Omega_{M,N}\times(k,\ell)}, \quad k,\ell \in [r] \tag{2}$$

to fully determine the kriging component of $\mathbf{X}|\mathbf{X}_{\mathrm{LR}}$. Remark that the matrix $\mathbf{\Lambda}$ is determined by MN values, the size of the HR image.

3.3. Resolution of the systems

To determine the kriging component, it is sufficient to solve the r^2 systems of Equation (2). We will use the Lemma 1.

Lemma 1 (Convolution and subsampling)**.** *If* $\mathbf{B}$ *is a 2D convolution on* $\Omega_{M,N}$ *by the kernel* $\boldsymbol{\beta} \in \Omega_{M,N}$, $\mathbf{SBS}^T$ *is a convolution on* $\Omega_{M/r,N/r}$ *by the kernel* $\mathbf{S}\boldsymbol{\beta} \in \Omega_{M/r,N/r}$.

By Lemma 1, $\mathbf{A\Gamma A}^T$ is a convolution matrix with kernel $\boldsymbol{\kappa} = \mathbf{S}(\mathbf{c} \star \boldsymbol{\gamma} \star \check{\mathbf{c}})$ where $\mathbf{c}$ is the kernel of $\mathbf{C}$. The Equations (2) become

$$\boldsymbol{\kappa} \star \boldsymbol{\lambda}(k,\ell) = \mathbf{A\Gamma}_{\Omega_{M,N}\times(k,\ell)}, \quad k,\ell \in [r] \tag{3}$$

and we can easily pseudo-invert the convolution by component-wise division in the Fourier domain.

The full FFT-based procedure is given by Algorithm 1. Note that to avoid artefacts due to non-periodicity of the images, as classically done we use the periodic plus smooth decomposition [19] to make the spectral computations. To work with non-zero mean images, the mean grayscale of the LR image should be subtracted at the beginning of the algorithm and added at the end. For RGB color images, the operations should be led on each channel. Note that the Gaussian noise $\mathbf{W}$ to generate the ADSN textures should be the same for each channel [16], as done for the results prensented in Figure 1.

4. SUPER-RESOLUTION WITH A REFERENCE IMAGE

In the previous theoretical sections we used the ground truth HR version $\mathbf{U}$ to estimate the distribution of the Gaussian texture, making the process impractical. In this section we demonstrate that the approach extends to the context of SR with a reference image, the reference image being used in place of the ground truth for modeling the ADSN distribution. Given an LR texture $\mathbf{U}_{\mathrm{LR}} = \mathbf{AU} \in \mathbb{R}^{\Omega_{M/r,N/r}}$ and a reference HR texture $\mathbf{U}_{\mathrm{ref}} \in \mathbb{R}^{\Omega_{M,N}}$, we simply replace the unkown ADSN kernel $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ by $\mathbf{t}_{\mathrm{ref}} = \frac{1}{\sqrt{MN}}(\mathbf{U}_{\mathrm{ref}} - m_{\mathrm{ref}})$ in Algorithm 1, assuming that the two microtextures are similar. We sample $\mathbf{X}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}} + (\tilde{\mathbf{X}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{X}})$ where $\tilde{\mathbf{X}}$ has the distribution $\mathrm{ADSN}(\mathbf{U}_{\mathrm{ref}})$ and $\mathbf{\Lambda}$ is the kriging matrix associated with $\mathrm{ADSN}(\mathbf{U}_{\mathrm{ref}})$.

Two SR experiments are given in Figure 2 where we evaluate and compare our results with two competitive methods: Wasserstein patch prior (WPP) [13] and SRFlow [8]. WPP[1] is deterministic and proposes to use optimal transport in patch space in a variational formulation for SR. The routine needs a reference image similar to ours. SRFlow[2] is a normalizing flow for sotchastic SR trained on natural images. The network transforms a standard Gaussian latent variable into a SR sample given the LR image. It depends on a temperature hyperparameter $\tau \in [0,1]$ that modulates the variance of the latent variable. Both methods require GPU and experiments were conducted using a NVIDIA V100 GPU. Visually, there are some issues in the borders of the image generated by WPP and the output texture tends to be blurry. The results of SRFlow with high temperature are too sharp for the Gaussian texture input, probably a bias induced by the training set of natural images that are often piecewise regular. The images obtained with SR Gaussian (ours) have a more granular appearence, due to the innovation component. However, the texture of details that breaks the stationary assumptions are not retrieved as in the white spot in the second image. Let us stress here that both WPP and SRFlow are designed for generic natural images while our method only works on Gaussian textures.

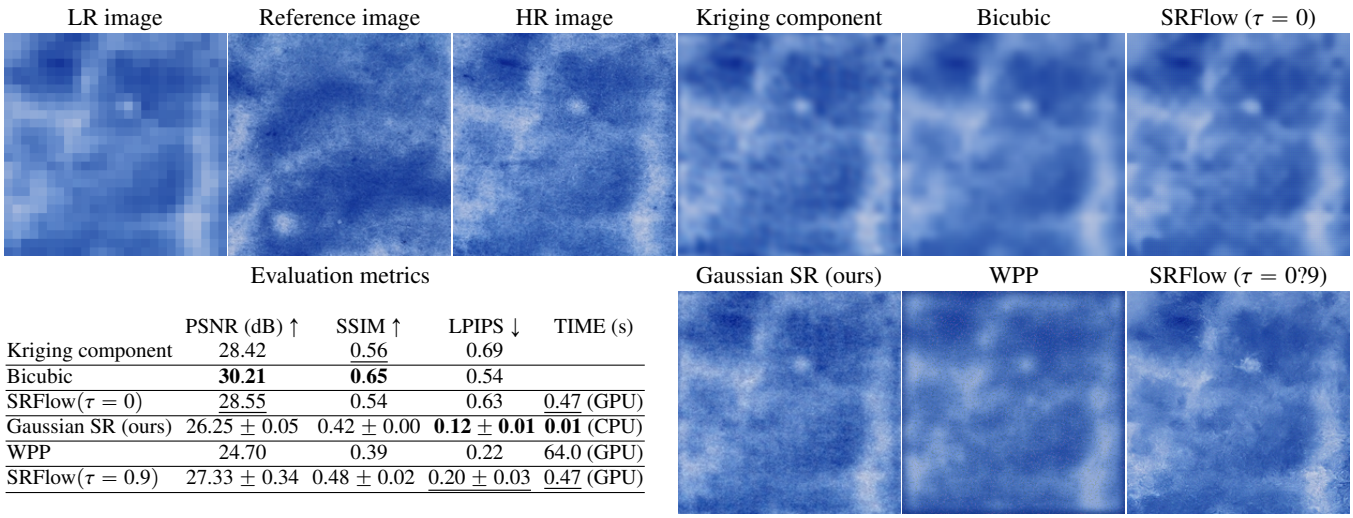
To evaluate the different methods we report in Figure 2 the metrics Peak Signal to Noise Ratio (PSNR), Structural SIMilarity (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). PSNR is the logarithm of the mean-squared error (MSE) between two images. SSIM is a metric which quantifies the similarity of two images studying their luminance, contrast and structure [20]. LPIPS is the norm between features of a pre-trained classification network and quantifies the perceptual similarity between two images [21]. As illustrated by the results of Figure 2, PSNR and SSIM are not adapted to evaluate the quality of the texture samples in our context. Indeed the best solutions for this metric are always the most blurry solutions, namely the bicubic interpolation, SRFlow with temperature $\tau = 0$, and the kriging component. The LPIPS metric is more relevant for SR of textures. For this metric our Gaussian SR results are the best for both examples. Note also that our algorithm runs very quickly using only a CPU.

To advocate for the irrelevance of the PSNR for stochastic texture SR, in our Gaussian framework it can be shown that the MSE is lower for the blurry kriging component than for the perfect SR samples.

Proposition 2 (Kriging component and MSE)**.** *Let* $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ *be a Gaussian texture,* $\mathbf{U}_{\mathrm{LR}} = \mathbf{AU}$ *its LR version,* $\mathbf{U}_{\mathrm{ref}}$ *a reference image,*

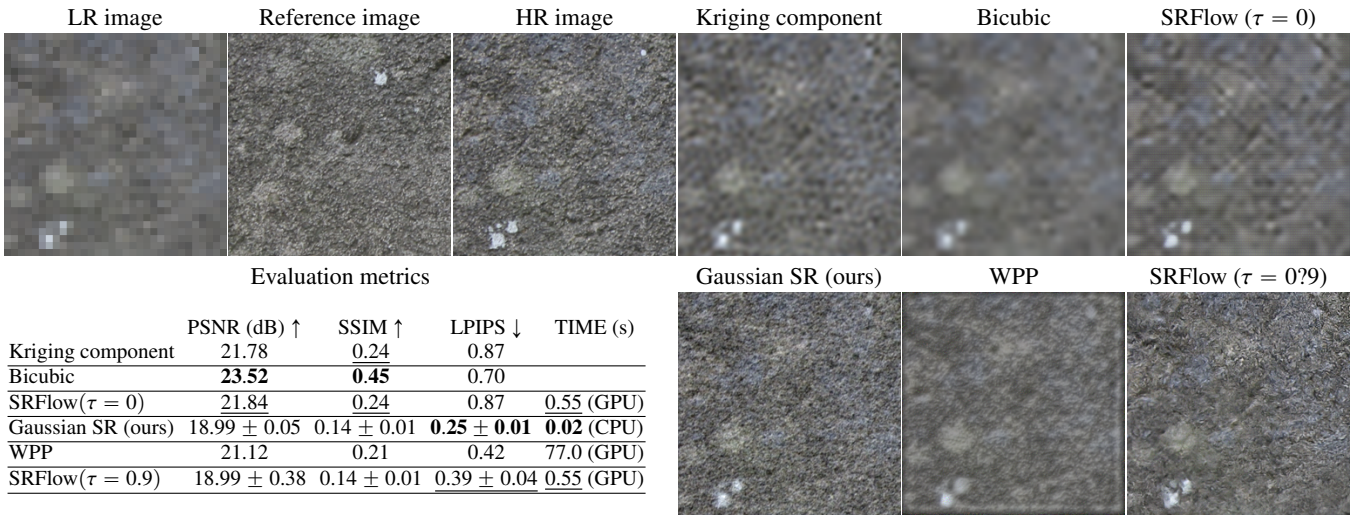
[1] `https://github.com/johertrich/Wasserstein_Patch_Prior`

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



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Kriging component	28.42	0.56	0.69	
Bicubic	**30.21**	**0.65**	0.54	
SRFlow($\tau = 0$)	28.55	0.54	0.63	0.47 (GPU)
Gaussian SR (ours)	26.25 ± 0.05	0.42 ± 0.00	**0.12 ± 0.01**	**0.01** (CPU)
WPP	24.70	0.39	0.22	64.0 (GPU)
SRFlow($\tau = 0.9$)	27.33 ± 0.34	0.48 ± 0.02	0.20 ± 0.03	0.47 (GPU)



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Kriging component	21.78	0.24	0.87	
Bicubic	**23.52**	**0.45**	0.70	
SRFlow($\tau = 0$)	21.84	0.24	0.87	0.55 (GPU)
Gaussian SR (ours)	18.99 ± 0.05	0.14 ± 0.01	**0.25 ± 0.01**	**0.02** (CPU)
WPP	21.12	0.21	0.42	77.0 (GPU)
SRFlow($\tau = 0.9$)	18.99 ± 0.38	0.14 ± 0.01	0.39 ± 0.04	0.55 (GPU)

Fig. 2. Gaussian SR with a reference image for a factor $\times 8$ and comparison with other methods. Top: HR size 208×208. Bottom: HR size 256×256. For the stochastic methods SR Gaussian and SRFlow, the table has been realized on 200 samples. Note that our method outperforms in terms of the perceptual LPIPS metric and execution time, while PSNR and SSIM are optimal for blurry images, illustrating that these metrics are not relevant for texture SR.

$\mathbf{\Gamma}$ *such that* $\mathrm{ADSN}(\mathbf{U}_{\mathrm{ref}}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ *and* $\mathbf{\Lambda}$ *associated to* $\mathbf{\Gamma}$. *Let* $\mathbf{X}_{\mathrm{SR}}$ *be a random image following the distribution of the SR samples obtained with Algorithm 1, then*

$$\begin{aligned}\mathbb{E}_{\mathbf{X}_{\mathrm{SR}}}\left(\|\mathbf{U} - \mathbf{X}_{\mathrm{SR}}\|^2\right) &= \|\mathbf{U} - \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}}\|^2 + \mathrm{Tr}(\mathbf{\Gamma} - \mathbf{\Lambda}^T\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda}) \\ &\geqslant \|\mathbf{U} - \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}}\|^2.\end{aligned}$$

In other words, the kriging component $\mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}}$ *has a lower MSE in expectation than the perfect SR samples* $\mathbf{X}_{\mathrm{SR}}$.

The inequality is due to the positiveness of the covariance matrix $\mathbf{\Gamma} - \mathbf{\Lambda}^T\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda}$ of the innovation component. This property is consistent with the fact that computing $\mathbb{E}(\mathbf{X}|\mathbf{X}_{\mathrm{LR}})$ aims to minimize the mean squared error. This is a specific illustration of the well-known regression to the mean problem in SR [22]. The same effect can be observed for the network SRFlow with temperature $\tau = 0$ which solves deterministically the MSE problem [8].

5. CONCLUSION

This study solves the problem of stochastic SR for stationary Gaussian textures. Such texture models constitute a base case for stochastic SR for which all the computations are accessible, leading to an efficient algorithm for stochastic SR and stochastic SR with a reference image. Our method outperforms some state of the art methods in terms of execution time and LPIPS metrics and we have demonstrated experimentally that the PSNR and the SSIM metrics are not compatible with the evaluation of the perceptual quality of SR samples. This illustrates the necessity to study further the performance on microtextures of generic stochastic SR models. Furthermore, the presented method could be applied for other inverse problems involving non invertible operators of the form convolution followed by subsampling.

Acknowledgements: The authors acknowledge the support of the project MISTIC (ANR-19-CE40-005).

6. REFERENCES

[1] Isola Phillip, Zhu Jun-Yan, Zhou Tinghui, Efros Alexei, "Image-to-image translation with conditional adversarial networks.," *Proceedings of the IEEE conference on computer vision and pattern recognition.*, 2017.

[2] Bruna, Joan, Sprechmann, Pablo, LeCun, Yann, "Super-resolution with deep convolutional sufficient statistics," *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[3] Ledig Christian, Theis Lucas, Huszár Ferenc, Caballero Jose, Cunningham Andrew, Acosta Alejandro, Aitken Andrew, Tejani Alykhan, Totz Johannes, Wang Zehan, and Shi Wenzhe, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114.

[4] 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 *Computer Vision – ECCV 2018 Workshops*, Laura Leal-Taixé and Stefan Roth, Eds., Cham, 2019, pp. 63–79, Springer International Publishing.

[5] Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Courville, Aaron Sherjil, Bengio Yoshua, "Generative adversarial nets.," *Advances in neural information processing systems.*, 2014.

[6] Johnson Justin, Alahi Alexandre, Fei-Fei Li, "Perceptual losses for real-time style transfer and super-resolution.," *European conference on computer vision.*, pp. 694–711, 2016.

[7] Simonyan Karen and Zisserman Andrew, "Very deep convolutional networks for large-scale image recognition," 2014.

[8] Lugmayr Andreas, Danelljan Martin, Van Gool Luc, and Timofte Radu, "Srflow: Learning the super-resolution space with normalizing flow," in *Computer Vision – ECCV 2020*, Cham, 2020, pp. 715–732, Springer International Publishing.

[9] Kingma Durk P and Dhariwal Prafulla, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. 2018, vol. 31, pp. 10215–10224, Curran Associates, Inc.

[10] Saharia Chitwan, Ho Jonathan, Chan William, Salimans Tim, Fleet David J., and Norouzi Mohammad, "Image super-resolution via iterative refinement," 2021.

[11] Ho, Jonathan, Jain, Ajain, Abbeel, Pieter, "Denoising diffusion probabilistic models.," *Advances in neural information processing systems.*, 2020.

[12] Song Yang and Ermon Stefano, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems*. 2019, vol. 32, Curran Associates, Inc.

[13] Hertrich Johannes, Houdard Antoine, and Redenbach Claudia, "Wasserstein patch prior for image superresolution," *IEEE Transactions on Computational Imaging*, vol. 8, pp. 693–704, 2022.

[14] Nguyen Dang-Phuong-Lan, Hertrich Johannes, Aujol Jean-Francois, and Berthoumieu Yannick, "Image super-resolution with PCA Reduced generalized Gaussian mixture models," working paper or preprint, May 2022.

[15] Chatillon Pierrick, Gousseau Yann, and Lefebvre Sidonie, "A statistically constrained internal method for single image super-resolution," in *International Conference on Pattern Recognition*, 2022.

[16] Galerne Bruno, Gousseau Yann, and Morel Jean-Michel, "Random phase textures: Theory and synthesis," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 257 – 267, 2011.

[17] Galerne Bruno, Leclaire Arthur, and Moisan Lionel, "Micro-texture inpainting through Gaussian conditional simulation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 1204–1208.

[18] Galerne Bruno and Leclaire Arthur, "Texture inpainting using efficient Gaussian conditional simulation," *SIAM Journal on Imaging Sciences*, vol. 10, no. 3, pp. 1446–1474, 2017.

[19] Moisan Lionel, "Periodic plus smooth image decomposition," *Journal of Mathematical Imaging and Vision*, vol. 39, no. 2, pp. 161–179, 2011.

[20] Wang Zhou, Bovikand Alan Conrad, Sheikh Hamid Rahim, and Simoncelli Eero P., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[21] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *CoRR*, vol. abs/1801.03924, 2018.

[22] Sønderby Casper Kaae, Caballero Jose, Theis Lucas, Shi Wenzhe, and Huszár Ferenc, "Amortised map inference for image super-resolution," in *International Conference on Learning Representations*, 2017.