Stochastic super-resolution for Gaussian textures

Émile Pierret[a], supervised by Bruno Galerne[a,b]

Séminaire des doctorants de LAREMA, June 27^{th}

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

Outline

A gentle introduction to images and textures

What is an image ?

$\in \mathbb{R}^{M \times N}$

Grayscale image

What is a RGB image ?

image $=$

$\in \mathbb{R}^{3\times M\times N}$

R

$\in \mathbb{R}^{1\times M\times N}$

G

$\in \mathbb{R}^{1\times M\times N}$

B

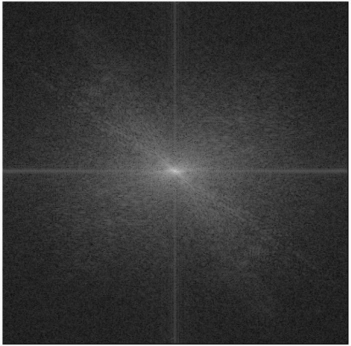
$\in \mathbb{R}^{1\times M\times N}$

Discrete Fourier transform

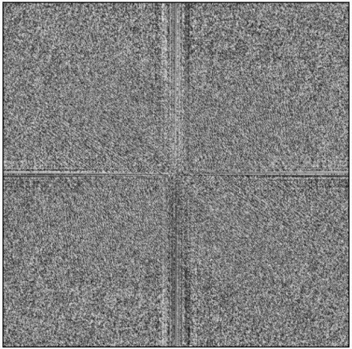
Let $\mathbf{X} \in \mathbb{R}^{M\times N}$ be an image. For $\xi = (\xi_1, \xi_2) \in [0, M-1] \times [0, N-1]$, we consider:

$$\widehat{\mathbf{X}}(\xi) = \sum_{(k,\ell)\in[0,M-1]\times[0,N-1]^2} \mathbf{X}(k,\ell) e^{-\frac{2ik\xi_1\pi}{M}} e^{-\frac{2i\ell\xi_2\pi}{N}}$$

image



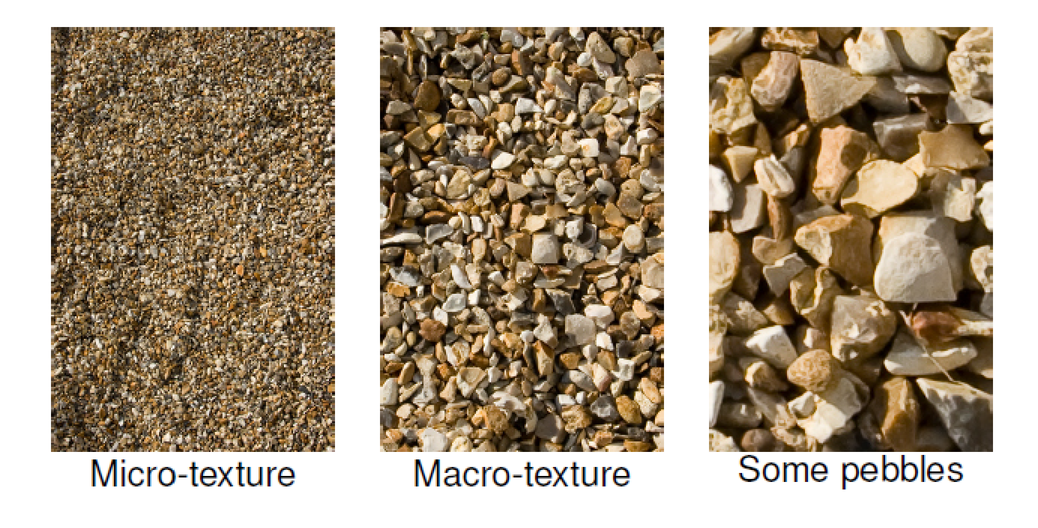
Modulus



Phase

Let $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{M \times N}$ be two images,

- FFT:
  $\hat{\mathbf{X}}$ can be computed in $\mathcal{O}(MN \log(MN))$ with the Fast Fourier Transform algorithm.
- Convolution:
  For $\boldsymbol{x} \in [0, M-1] \times [0, N-1]$, if $(\mathbf{X} \star \mathbf{Y})(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in [0, n-1]^2} \mathbf{X}\big(\boldsymbol{x} - \boldsymbol{y}\big)\mathbf{Y}(\boldsymbol{y})$,
  $$\widehat{\mathbf{X} \star \mathbf{Y}} = \hat{\mathbf{X}} \odot \hat{\mathbf{Y}}$$



Images extracted from Bruno Galerne's slides

Introduction to stochastic super-resolution

LR image (62×47 pixels)

Image extracted from dark sources

The number of pixels has been divided by $r = 32$.

The super-resolution

LR image (62×47 pixels)

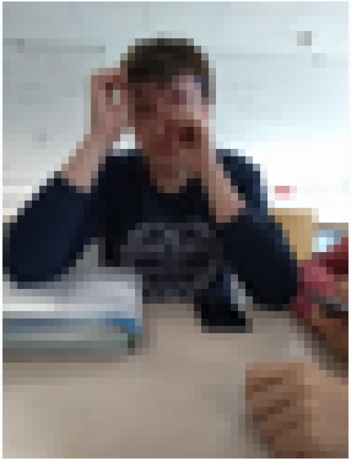
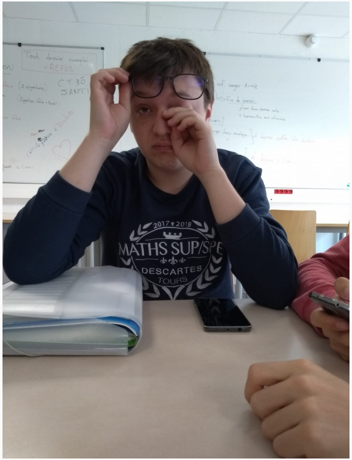


Image extracted from dark sources

The number of pixels has been divided by $r = 32$.

The super-resolution

HR image (1984×1504 pixels)



LR image (62×47 pixels)

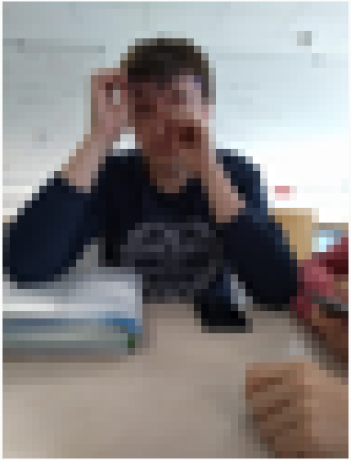
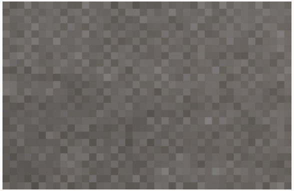


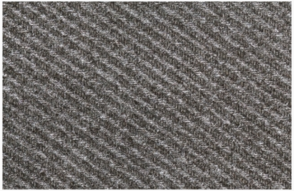
Image extracted from dark sources

The number of pixels has been divided by $r = 32$.



$= \mathbf{S}$

$= \underbrace{\mathbf{SC}}_{=\mathbf{A}}$



LR image — convoluted image — HR image

- SISR setting litterature: [Bruna et al., 2016][1] [Ledig et al., 2017][2], [Wang et al., 2019][3], [Johnson et al., 2016][4], [Hertrich, Houdard, et al., 2022][5], [Hertrich, Nguyen, et al., 2022][6], [Chatillon et al., 2022][7]

[1] Bruna, J., Sprechmann, P., & LeCun, Y. (2016). Super-Resolution with Deep Convolutional Sufficient Statistics. *ICLR 2016*

[2] Ledig, C., Theis, L., & Huszár, F. e. a. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *CVPR 2017*

[3] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2019). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *ECCV 2018*

[4] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *ECCV 2016*

[5] Hertrich, J., Houdard, A., & Redenbach, C. (2022). Wasserstein Patch Prior for Image Superresolution. *IEEE Transactions on Computational Imaging*

[6] Hertrich, J., Nguyen, L. D. P., Aujol, J.-F., Bernard, D., Berthoumieu, Y., Saadaldin, A., & Steidl, G. (2022). PCA Reduced Gaussian Mixture Models with Applications in Superresolution. *Inverse Problems and Imaging*

[7] Chatillon, P., Gousseau, Y., & Lefebvre, S. (2022). A statistically constrained internal method for single image super-resolution. *ICPR 2022*

The stochastic super-resolution

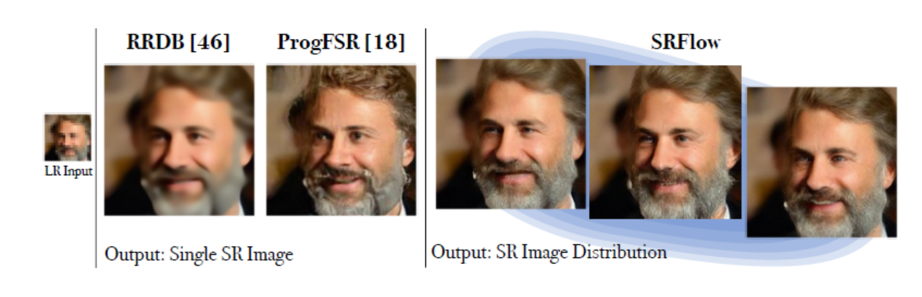


Image extracted from [Lugmayr et al., 2020]

Stochastic super-resolution litterature: SRFlow [Lugmayr et al., 2020][8], SR3 [Saharia et al., 2022][9], CEM [Bahat and Michaeli, 2020][10].

[8] Lugmayr, A., Danelljan, M., Van Gool, L., & Timofte, R. (2020). SRFlow: Learning the Super-Resolution Space with Normalizing Flow. *ECCV 2020*

[9] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image Super-Resolution Via Iterative Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[10] Bahat, Y., & Michaeli, T. (2020). Explorable super resolution. *CVPR*

Stochastic super-resolution: an example

HR image

LR image (by a factor $r = 16$)

Image of Robert Hossein extracted from the dataset CelebA.

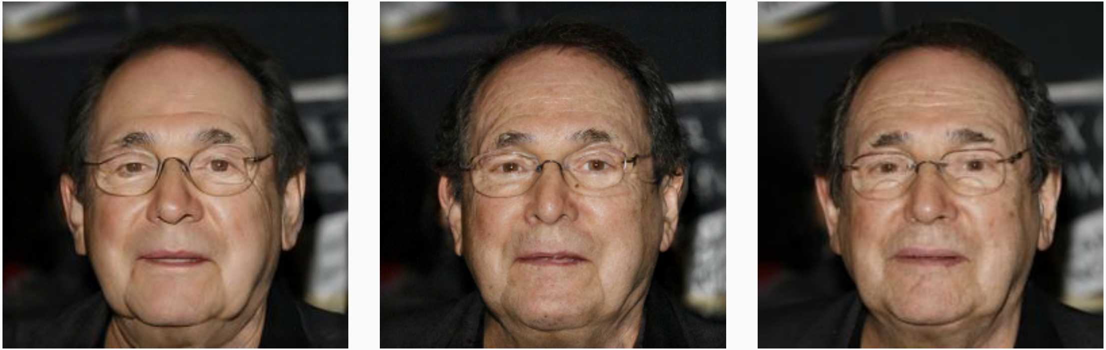


Image of Robert Hossein extracted from the dataset CelebA.

All these images have the same LR version !

Stochastic super-resolution: an example

Image of Robert Hossein extracted from the dataset CelebA.

Image extracted from [Galerne et al., 2011a] [11]

[11] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

Stochastic super-resolution for textures

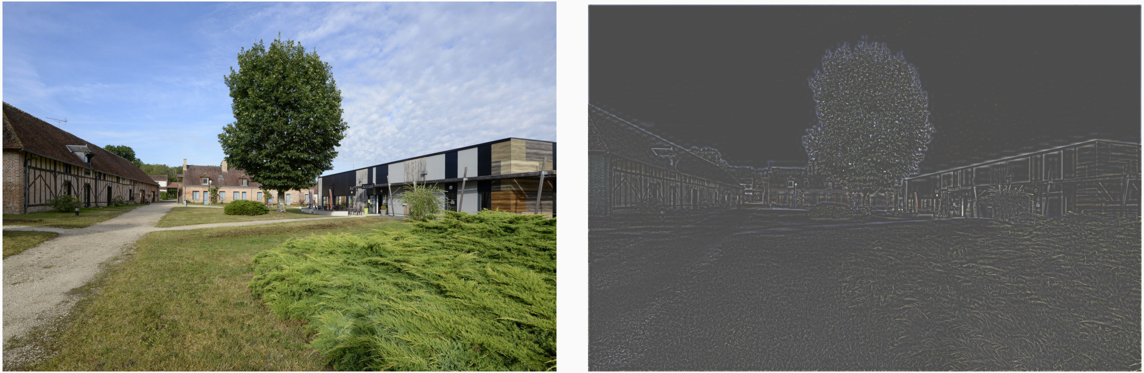
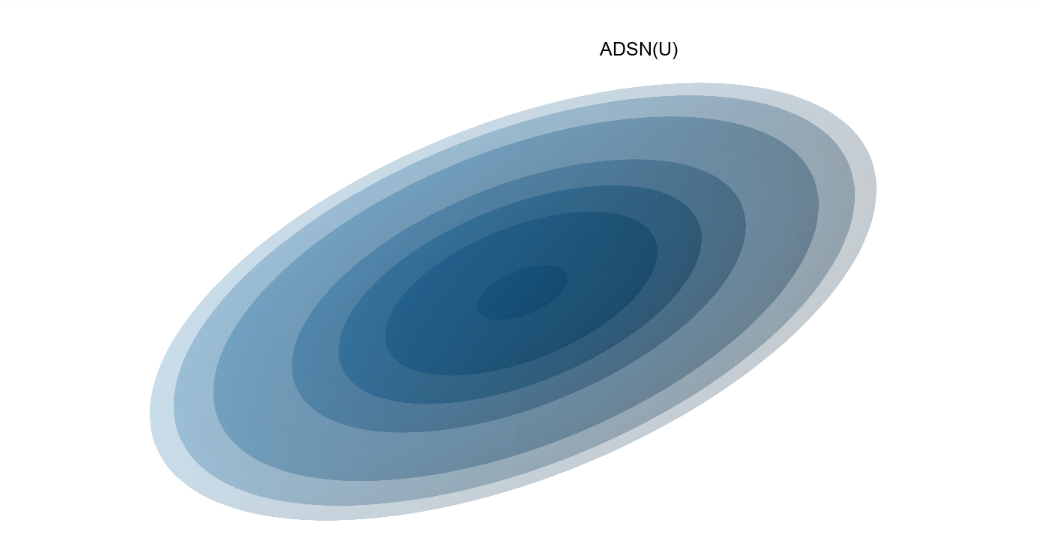
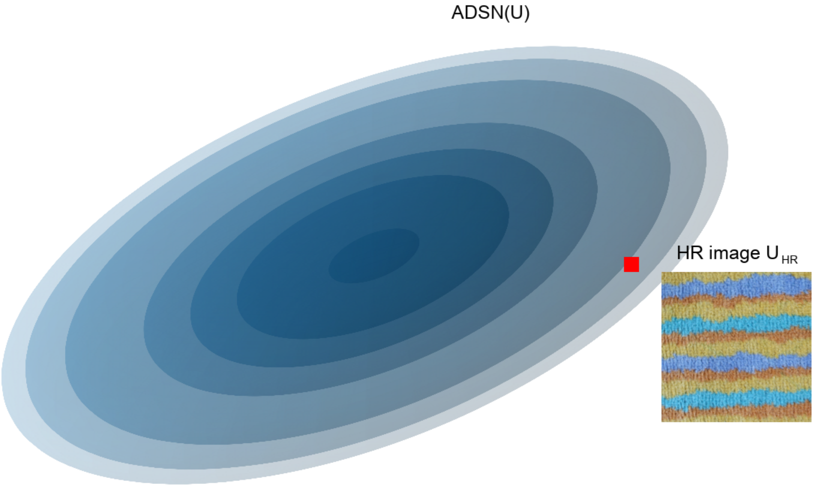
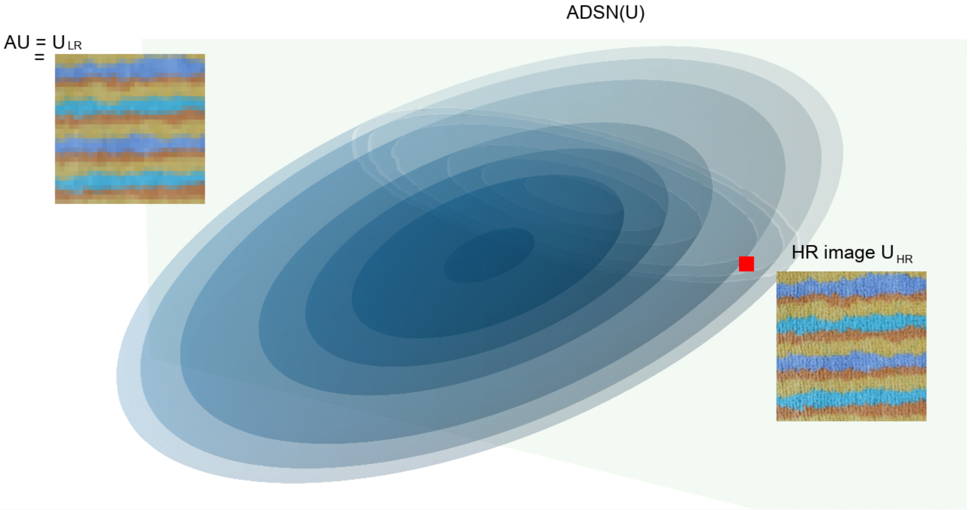


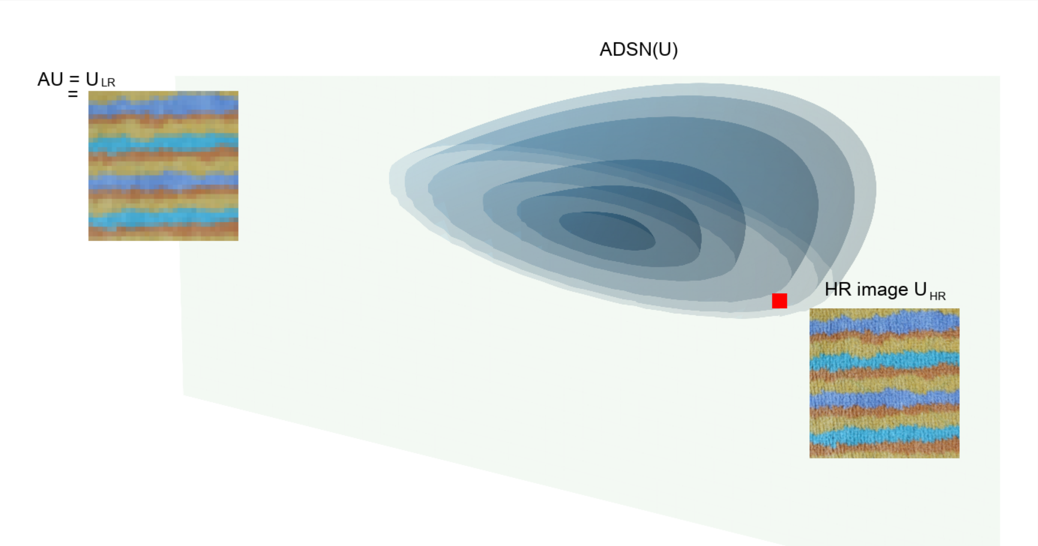
Image extracted from Courcimont website

The Gaussian SR method


ADSN(U)


ADSN(U)
HR image U HR


ADSN(U)
AU = U LR
=
HR image U HR


ADSN(U)
AU = U LR
=
HR image U HR

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [13]

Let $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{U}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma}) \qquad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored.

$\mathbf{U}$ Samples of $\mathrm{ADSN}(\mathbf{U})$

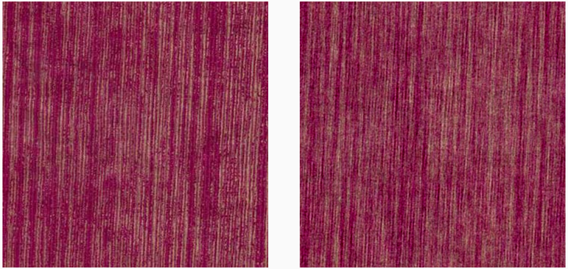


Image extracted from [Galerne et al., 2011a][12]

[12] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [13]

Let $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{U}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored.

$\mathbf{U}$

Samples of $\mathrm{ADSN}(\mathbf{U})$

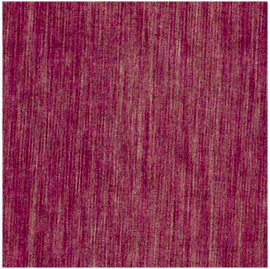


Image extracted from [Galerne et al., 2011a][12]

[12] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [13]

Let $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{U}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored.

$\mathbf{U}$

Samples of $\mathrm{ADSN}(\mathbf{U})$

Image extracted from [Galerne et al., 2011a][12]

[12] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [13]

Let $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{U}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored.

$\mathbf{U}$

Samples of $\mathrm{ADSN}(\mathbf{U})$

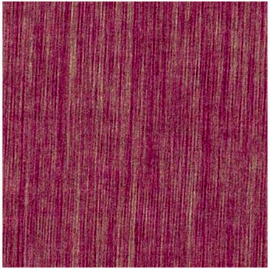


Image extracted from [Galerne et al., 2011a][12]

[12] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [13]

Let $\mathbf{U} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{U} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{U}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored.

$\mathbf{U}$ Samples of $\mathrm{ADSN}(\mathbf{U})$

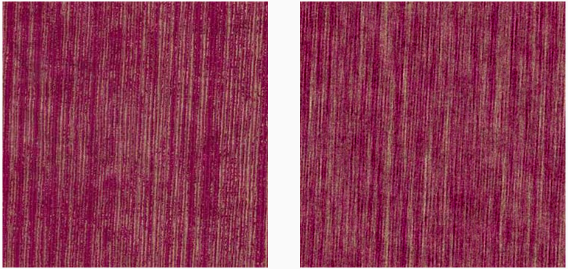


Image extracted from [Galerne et al., 2011a][12]

[12] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The conditional Gaussian simulation

Following ideas from [Galerne and Leclaire, 2017][14] 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:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) + [\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}})] \sim \mathbf{X}|\mathbf{AX}$$

[14] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

The conditional Gaussian simulation

Following ideas from [Galerne and Leclaire, 2017][14] Let $\mathbf{X}$ be a **Gaussian** vector, and $\mathbf{A}$ be a linear operator, $\mathbb{E}(\mathbf{X}|\mathbf{A}\mathbf{X})$ and $\mathbf{X} - \mathbb{E}(\mathbf{X}|\mathbf{A}\mathbf{X})$ are independent. Consequently, if $\tilde{\mathbf{X}}$ is independent of $\mathbf{X}$ with the same distribution, then:

$$\mathbb{E}(\mathbf{X}|\mathbf{A}\mathbf{X}) + [\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}})] \sim \mathbf{X}|\mathbf{A}\mathbf{X}$$

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 $\mathbb{E}(\mathbf{X}|\mathbf{A}\mathbf{X}) = \mathbf{\Lambda}^T \mathbf{A}\mathbf{X}$ and:

$$\mathbb{E}(\mathbf{X}|\mathbf{A}\mathbf{X}) = \mathbf{\Lambda}^T \mathbf{A}\mathbf{X} \iff \mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}.$$

[14] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

Following ideas from [Galerne and Leclaire, 2017][14] 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:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) + [\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}})] \sim \mathbf{X}|\mathbf{AX}$$

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:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T \mathbf{AX} \iff \mathbf{A\Gamma A}^T \mathbf{\Lambda} = \mathbf{A\Gamma}.$$

Consequence: To sample $\mathbf{X}_{\mathrm{SR}} \sim \mathrm{ADSN}(\mathbf{U}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$, conditioned on $\mathbf{AX}_{\mathrm{SR}} = \mathbf{U}_{\mathrm{LR}}$, we aim:

$$\mathbf{\Lambda}^T \mathbf{U}_{\mathrm{LR}} + (\tilde{\mathbf{X}} - \mathbf{\Lambda}^T \mathbf{A}\tilde{\mathbf{X}}) \quad \text{with } \tilde{\mathbf{X}} \sim \mathrm{ADSN}(\mathbf{U}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$$

[14] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

The Gaussian SR method

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

$$\mathbf{X}_{\mathrm{SR}} = \mathbf{\Lambda}^T \mathbf{U}_{\mathrm{LR}} + (\tilde{\mathbf{X}} - \mathbf{\Lambda}^T \mathbf{A}\tilde{\mathbf{X}}) \quad \text{with } \tilde{\mathbf{X}} \sim \mathrm{ADSN}(\mathbf{U}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}), \text{ independent of } \mathbf{U}_{\mathrm{HR}}$$

With $\mathbf{\Lambda}$ verifying the kriging equation:

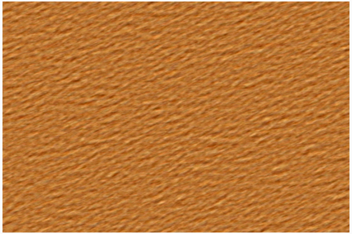
$$\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T \mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}. \tag{1}$$

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

$=$

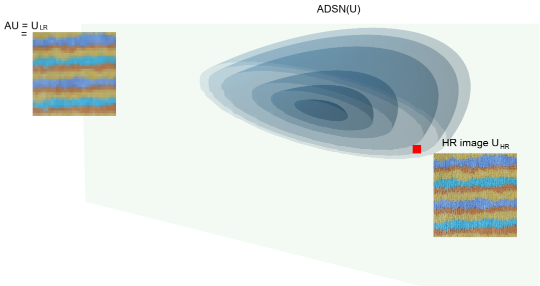
Kriging component: $\mathbf{\Lambda}^T \mathbf{U}_{\mathrm{LR}}$

$+$

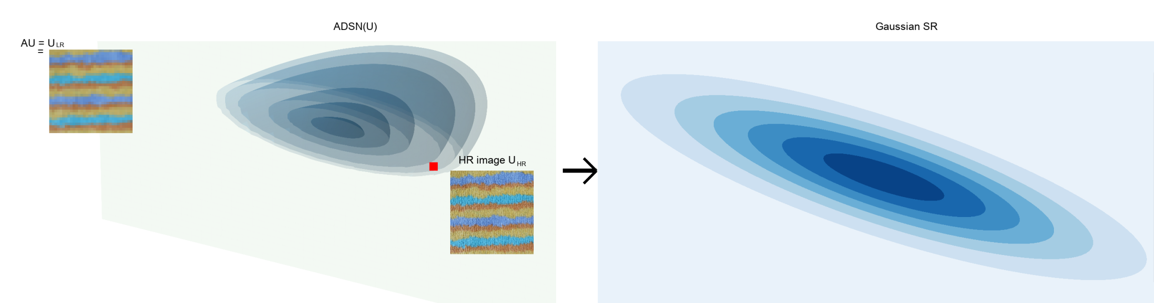


Innovation component: $\tilde{\mathbf{X}} - \mathbf{\Lambda}^T \mathbf{A}\tilde{\mathbf{X}}$

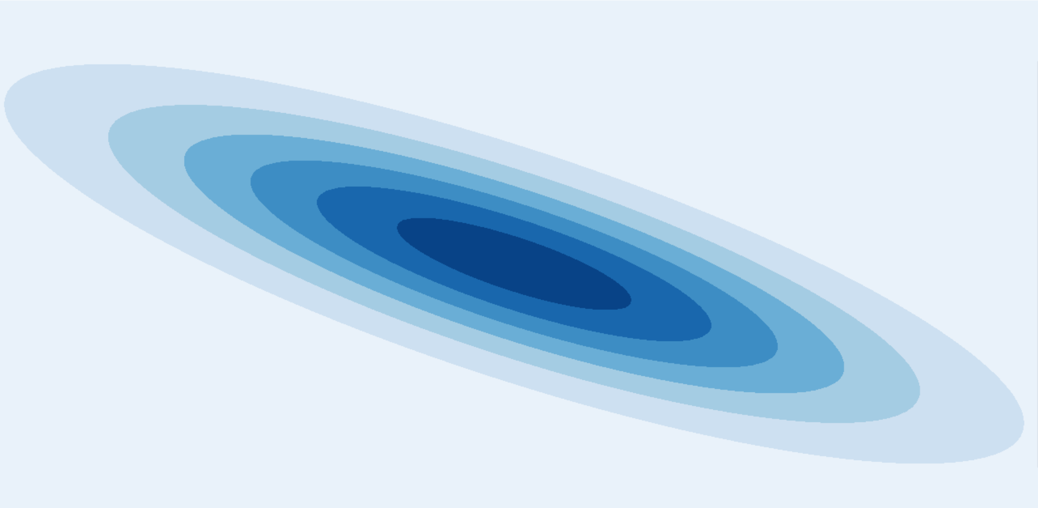
Image extracted from [Galerne et al., 2011a]



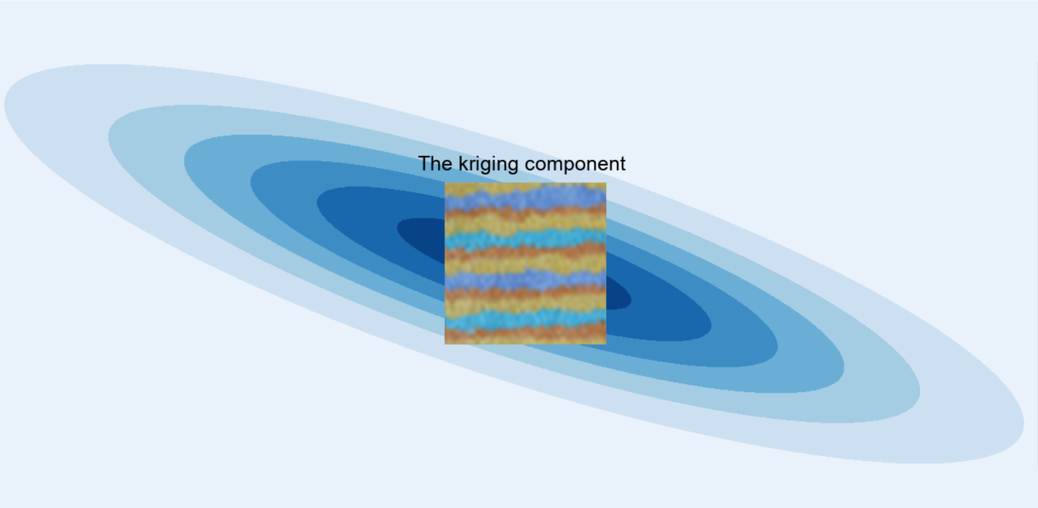
ADSN(U)
AU = U_LR
HR image U_HR


ADSN(U)
Gaussian SR
AU = U_LR
HR image U_HR

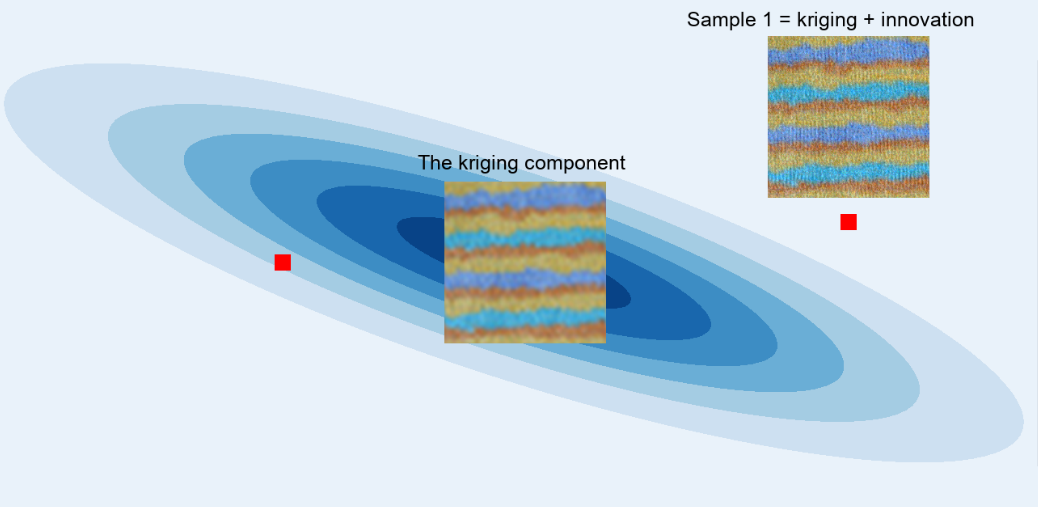
Gaussian SR



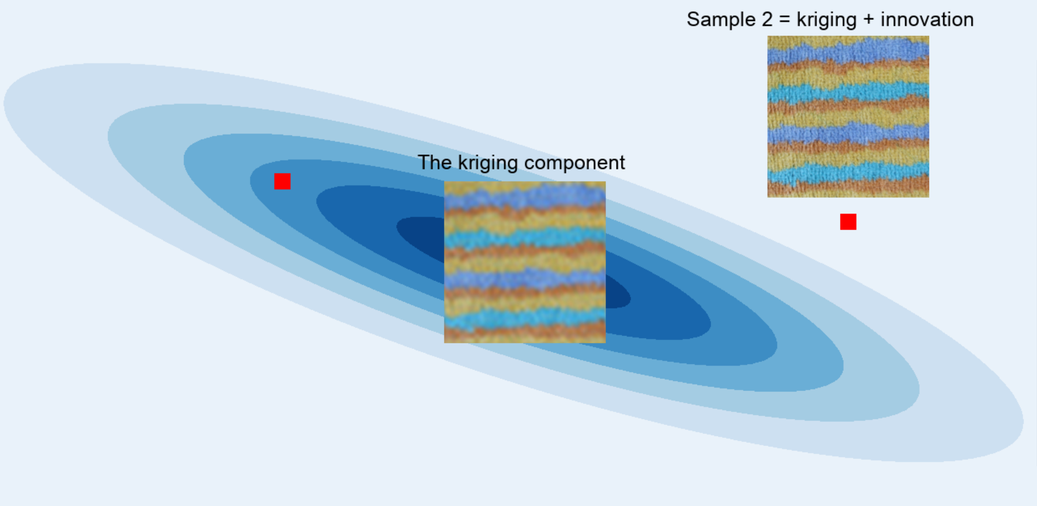
Gaussian SR



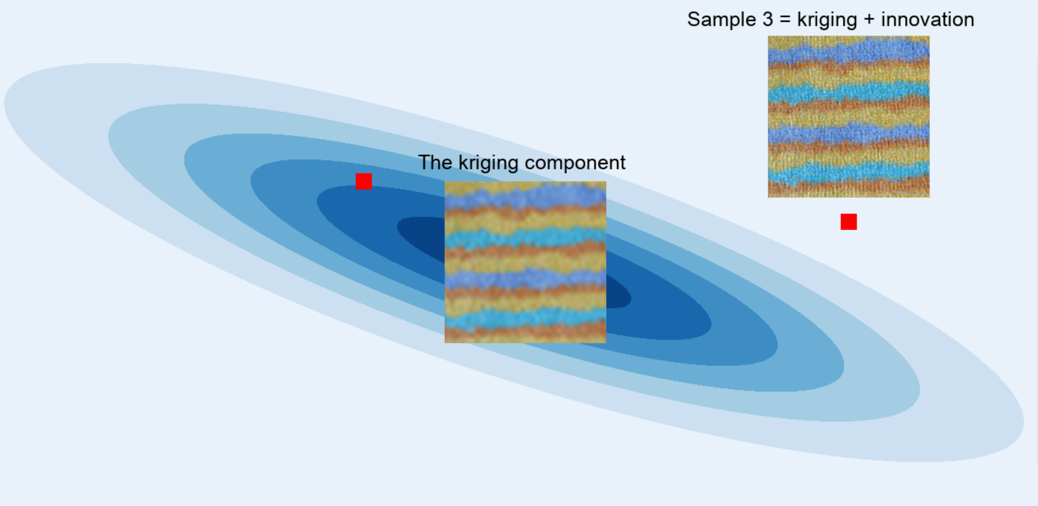
Gaussian SR



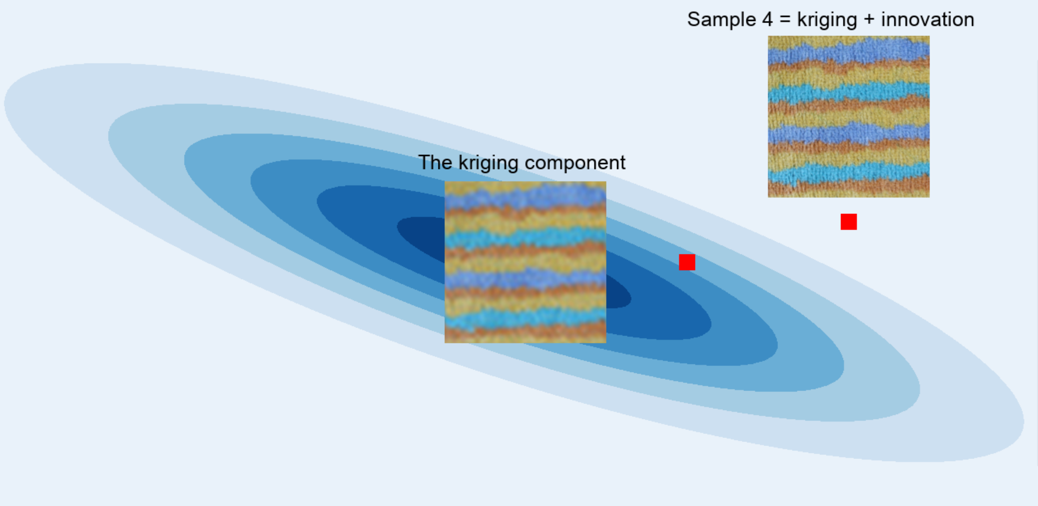
Gaussian SR



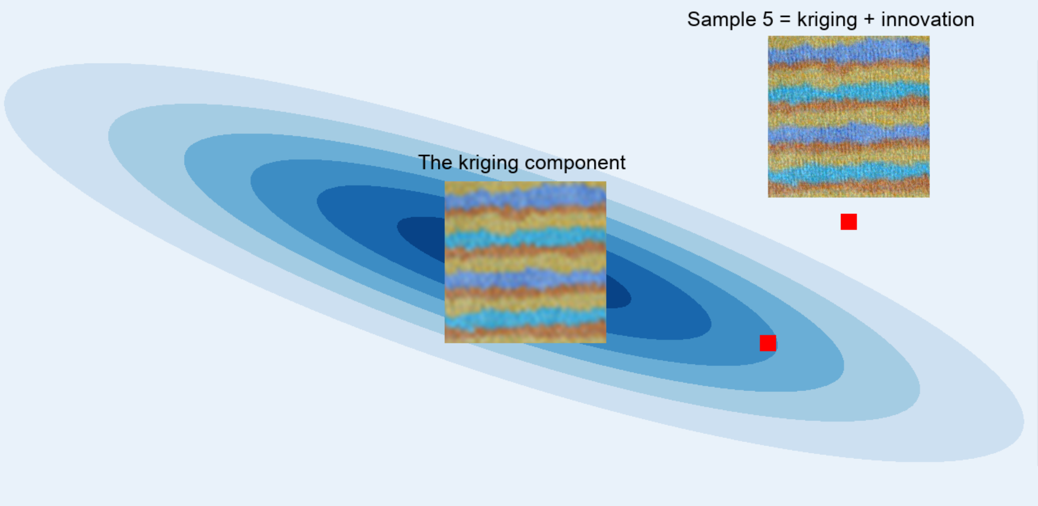
Gaussian SR



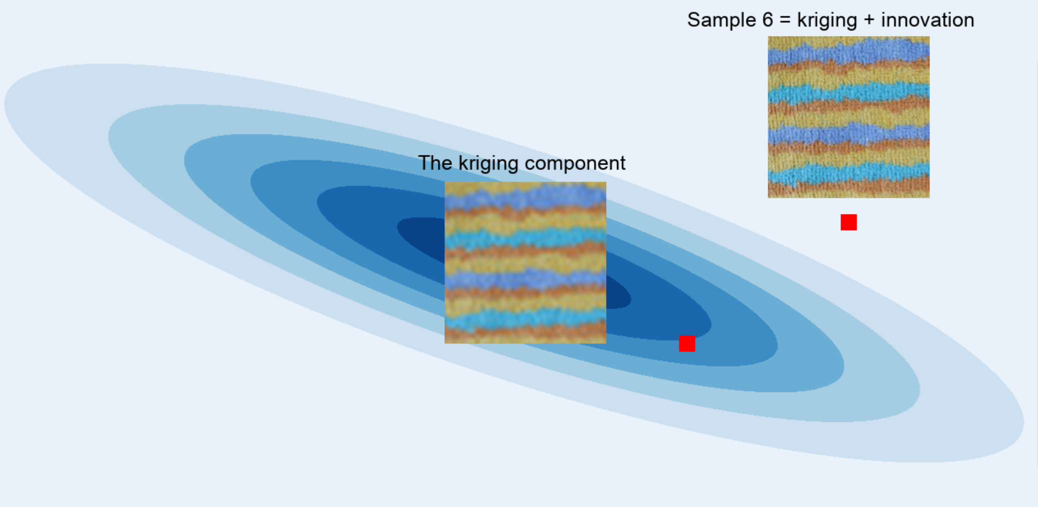
Gaussian SR



Gaussian SR



Gaussian SR



Solving the kriging equation

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

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

Problem 1 : How to solve this equation ?

Solving the kriging equation

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

Problem 1 : How to solve this equation ?

Problem 2 : $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ is very large. How to store it ?

Solving the kriging equation

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

Problem 1 : How to solve this equation ?

Problem 2 : $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r}\times\Omega_{M,N}}$ is very large. How to store it ?

Solution 1 : $\mathbf{A\Gamma A}^T$ is a convolution matrix. For each column $\boldsymbol{\lambda}(\boldsymbol{x})$ of $\mathbf{\Lambda}$, the equation becomes:

$$\kappa \star \boldsymbol{\lambda}(\boldsymbol{x}) = \mathbf{A\Gamma}_{\mathbb{R}^{n\times n}\times\{\boldsymbol{x}\}}$$

ie

$$\widehat{\kappa} \odot \widehat{\boldsymbol{\lambda}(\boldsymbol{x})} = \widehat{\mathbf{A\Gamma}}_{\mathbb{R}^{n\times n}\times\{\boldsymbol{x}\}}$$

Solving the kriging equation

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

Problem 1 : How to solve this equation ?

Problem 2 : $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ is very large. How to store it ?

Solution 1 : $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T$ is a convolution matrix. For each column $\boldsymbol{\lambda}(\boldsymbol{x})$ of $\mathbf{\Lambda}$, the equation becomes:

$$\kappa \star \boldsymbol{\lambda}(\boldsymbol{x}) = \mathbf{A}\mathbf{\Gamma}_{\mathbb{R}^{n \times n} \times \{\boldsymbol{x}\}}$$

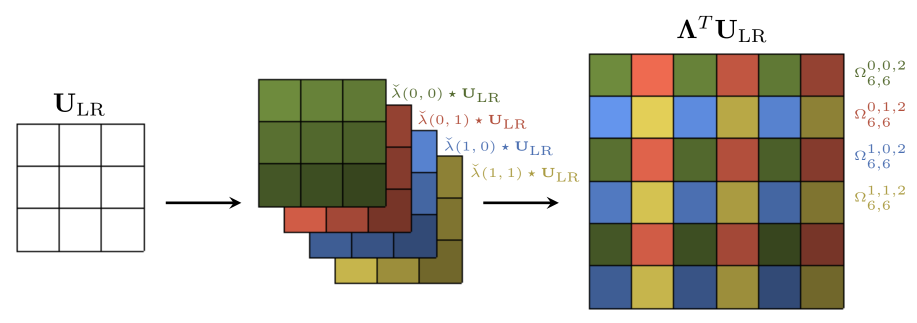
ie

$$\widehat{\kappa} \odot \widehat{\boldsymbol{\lambda}(\boldsymbol{x})} = \widehat{\mathbf{A}\mathbf{\Gamma}}_{\mathbb{R}^{n \times n} \times \{\boldsymbol{x}\}}$$

Solution 2 : There exists a matrix $\mathbf{\Lambda}$ such that it has a convolutional behavior. Only r^2 columns of $\mathbf{\Lambda}$ are necessary. To store this solution $\mathbf{\Lambda}$, only a HR size is necessary.

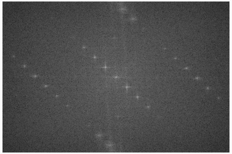
Application of Λ

For $r = 2$, HR size : 6×6, LR size : 3×3.

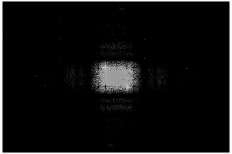


λ convolution

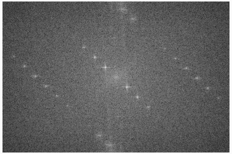
HR image



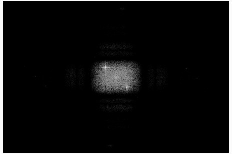
$\boldsymbol{\lambda}$



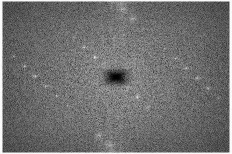
Sample



Kriging comp.



Innovation comp.



Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{U}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{U}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{U}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$
- We can simulate SR samples $\mathbf{X}_{\mathrm{SR}}$ compatible with $\mathbf{U}_{\mathrm{LR}}$ computing $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and taking $\mathbf{X}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}} + \tilde{\boldsymbol{Z}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\boldsymbol{Z}}$ with $\tilde{\boldsymbol{Z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{U}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$
- We can simulate SR samples $\mathbf{X}_{\mathrm{SR}}$ compatible with $\mathbf{U}_{\mathrm{LR}}$ computing $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and taking $\mathbf{X}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}} + \tilde{\boldsymbol{Z}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\boldsymbol{Z}}$ with $\tilde{\boldsymbol{Z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{U}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$
- We can simulate SR samples $\mathbf{X}_{\mathrm{SR}}$ compatible with $\mathbf{U}_{\mathrm{LR}}$ computing $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and taking $\mathbf{X}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}} + \tilde{\boldsymbol{Z}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\boldsymbol{Z}}$ with $\tilde{\boldsymbol{Z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$

Γ is extracted from $\mathbf{U}_{\mathrm{HR}}$

Summary

- We know $\boldsymbol{\Gamma}$ such that $\mathbf{U}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$
- We can easily compute and store $\boldsymbol{\Lambda}$ such that $\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Lambda} = \mathbf{A}\boldsymbol{\Gamma}$
- We can simulate SR samples $\mathbf{X}_{\mathrm{SR}}$ compatible with $\mathbf{U}_{\mathrm{LR}}$ computing $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and taking $\mathbf{X}_{\mathrm{SR}} = \boldsymbol{\Lambda}^T\mathbf{U}_{\mathrm{LR}} + \tilde{\boldsymbol{Z}} - \boldsymbol{\Lambda}^T\mathbf{A}\tilde{\boldsymbol{Z}}$ with $\tilde{\boldsymbol{Z}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$

Γ is extracted from $\mathbf{U}_{\mathrm{HR}}$

Solution : Use a reference image with the same law.

With reference image

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][15]

[15] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][15]

[15] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][15]

[15] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][15]

[15]Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

Reference image | HR image | Samples

Image extracted from [Galerne et al., 2011a][15]

[15] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][15]

[15] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][16]

[16] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, *1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][16]

[16] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

		
Reference image	HR image	Samples

Image extracted from [Galerne et al., 2011a][16]

[16] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][16]

[16] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][16]

[16] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, *1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image



Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][16]

[16] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, *1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][17]

[17] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn


Reference image


HR image


Samples

Image extracted from [Galerne et al., 2011a][17]

[17]Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

Reference image

HR image

Samples

Image extracted from [Galerne et al., 2011a][17]

[17] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][17]

[17] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image HR image Samples

Image extracted from [Galerne et al., 2011a][17]

[17] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, *1*. https://doi.org/10.5201/ipol.2011.ggm-rpn

With reference image

Reference image

HR image

Samples

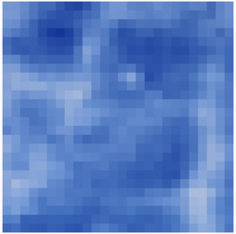
Image extracted from [Galerne et al., 2011a][17]

[17] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm-rpn

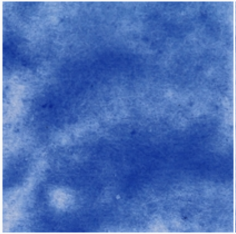
Comparison of Gaussian SR with other methods

Comparison with other methods

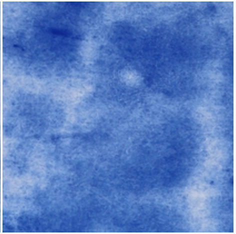
LR image



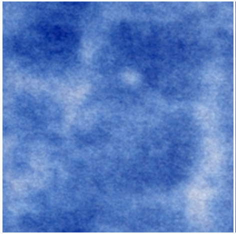
Reference image[18]



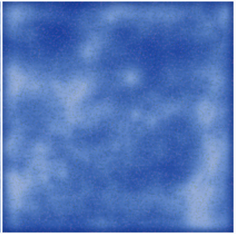
HR image



Gaussian SR (ours)



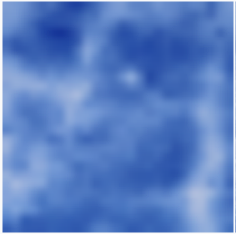
WPP



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**30.21**	**0.65**	0.54	
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)

Bicubic

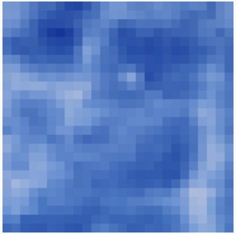


SRFlow ($\tau = 0.9$)

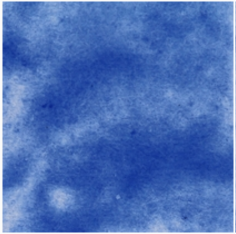
[18] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

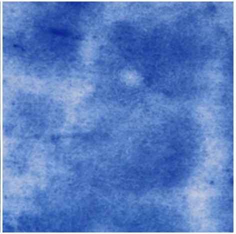
LR image



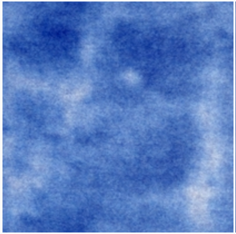
Reference image[18]



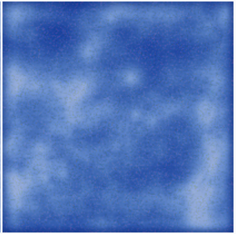
HR image



Gaussian SR (ours)



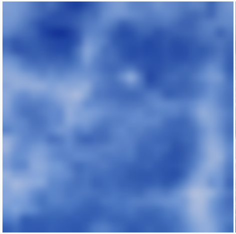
WPP



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**30.21**	**0.65**	0.54	
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)

Bicubic

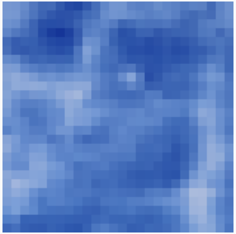


SRFlow ($\tau = 0.9$)

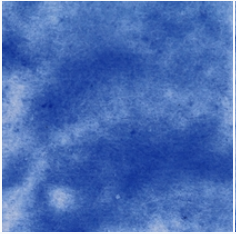
[18] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

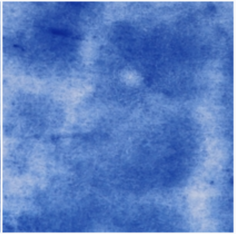
LR image



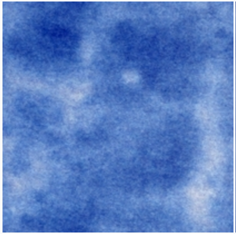
Reference image[18]



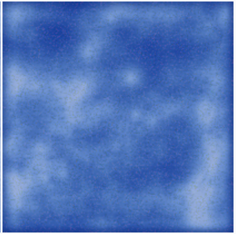
HR image



Gaussian SR (ours)



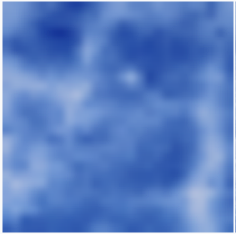
WPP



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**30.21**	**0.65**	0.54	
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	<u>0.20 ± 0.03</u>	<u>0.47</u> (GPU)

Bicubic

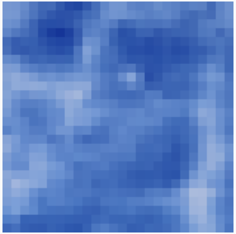


SRFlow ($\tau = 0.9$)

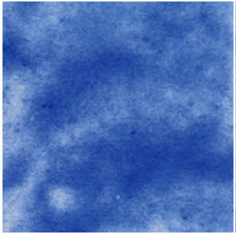
[18] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

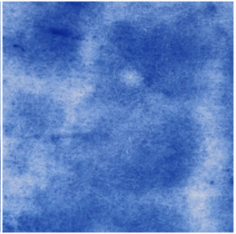
LR image



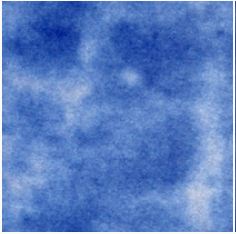
Reference image[18]



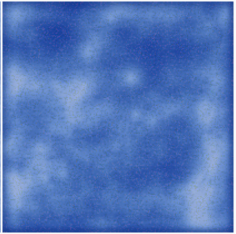
HR image



Gaussian SR (ours)



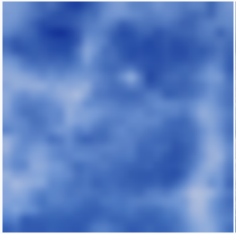
WPP



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**30.21**	**0.65**	0.54	
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	<u>0.20 ± 0.03</u>	<u>0.47</u> (GPU)

Bicubic

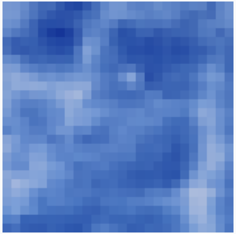


SRFlow ($\tau = 0.9$)

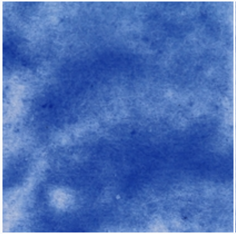
[18] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

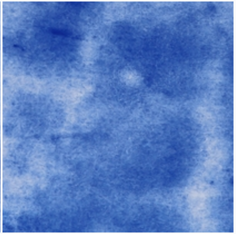
LR image



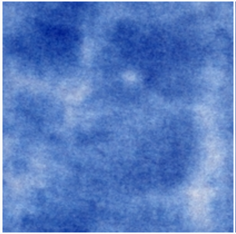
Reference image[18]



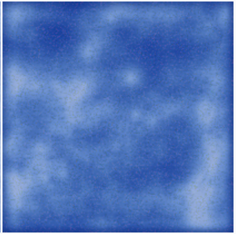
HR image



Gaussian SR (ours)



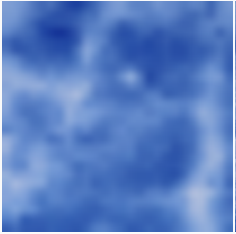
WPP



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**30.21**	**0.65**	0.54	
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	<u>0.20 ± 0.03</u>	<u>0.47</u> (GPU)

Bicubic

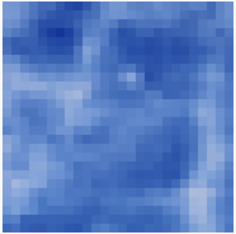


SRFlow ($\tau = 0.9$)

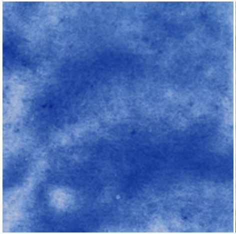
[18] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, 1.

Comparison with other methods

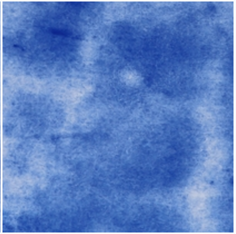
LR image



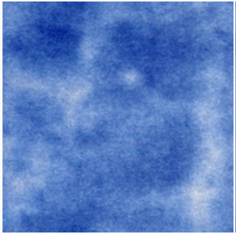
Reference image[18]



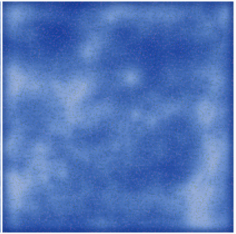
HR image



Gaussian SR (ours)



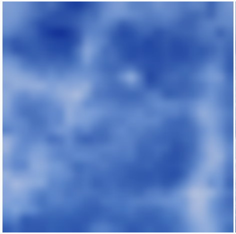
WPP



Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**30.21**	**0.65**	0.54	
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)

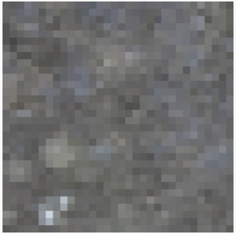
Bicubic



SRFlow ($\tau = 0.9$)

[18] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

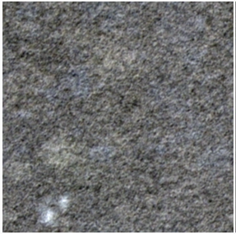
LR image



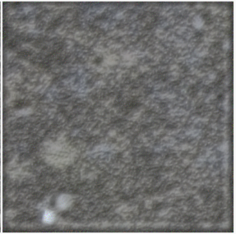
Reference image[19]

HR image

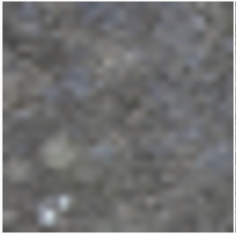
Gaussian SR (ours)



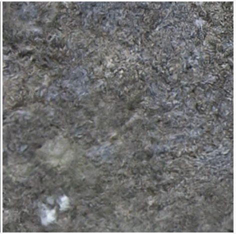
WPP



Bicubic



SRFlow ($\tau = 0.9$)



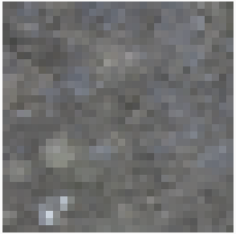
Evaluation metrics

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

[19] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

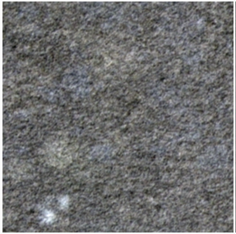
LR image



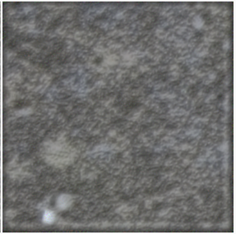
Reference image[19]

HR image

Gaussian SR (ours)



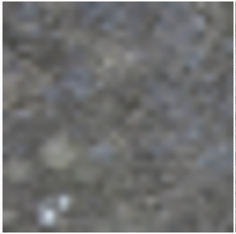
WPP



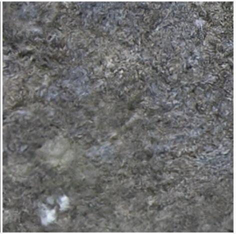
Evaluation metrics

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

Bicubic



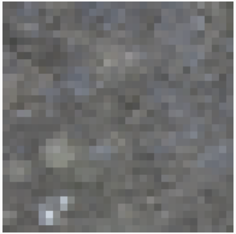
SRFlow ($\tau = 0.9$)



[19] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

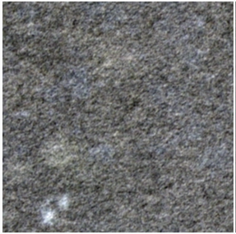
LR image



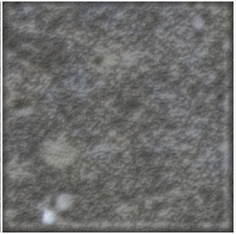
Reference image[19]

HR image

Gaussian SR (ours)



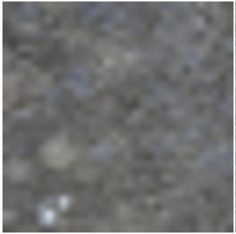
WPP



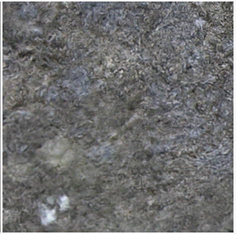
Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**23.52**	**0.45**	0.70	
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	<u>0.39 ± 0.04</u>	<u>0.55</u> (GPU)

Bicubic



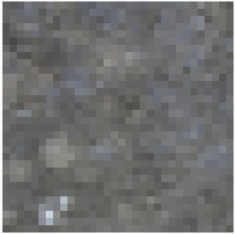
SRFlow ($\tau = 0.9$)



[19] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

Comparison with other methods

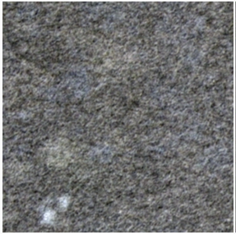
LR image



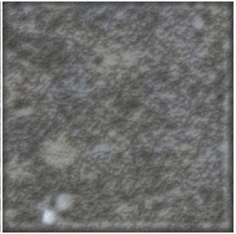
Reference image[19]

HR image

Gaussian SR (ours)



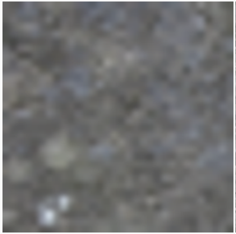
WPP



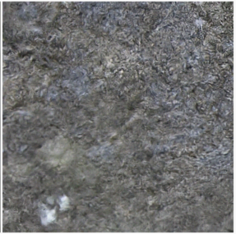
Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**23.52**	**0.45**	0.70	
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	<u>0.39 ± 0.04</u>	<u>0.55</u> (GPU)

Bicubic



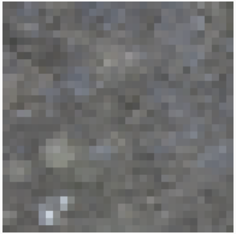
SRFlow ($\tau = 0.9$)



[19] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, 1.

Comparison with other methods

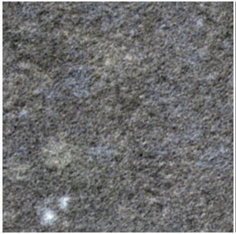
LR image



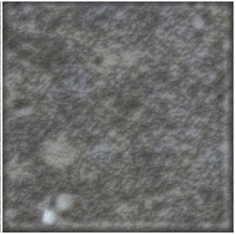
Reference image[19]

HR image

Gaussian SR (ours)



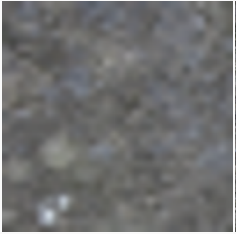
WPP



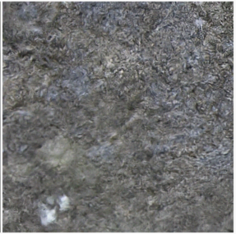
Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**23.52**	**0.45**	0.70	
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	<u>0.39 ± 0.04</u>	<u>0.55</u> (GPU)

Bicubic

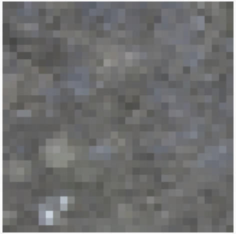


SRFlow ($\tau = 0.9$)



[19] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.*

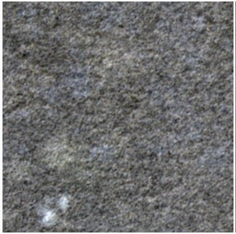
LR image



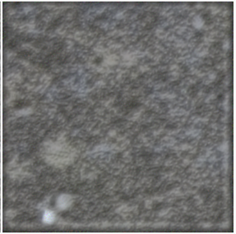
Reference image[19]

HR image

Gaussian SR (ours)



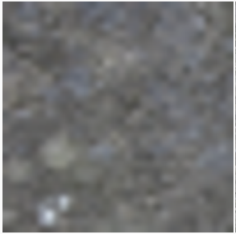
WPP



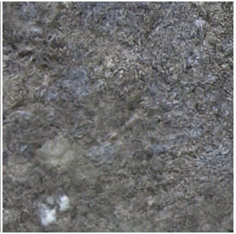
Evaluation metrics

	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	TIME (s)
Bicubic	**23.52**	**0.45**	0.70	
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)

Bicubic



SRFlow ($\tau = 0.9$)



[19] Images are extracted from Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, 1.

Proposition 1: Kriging component and MSE

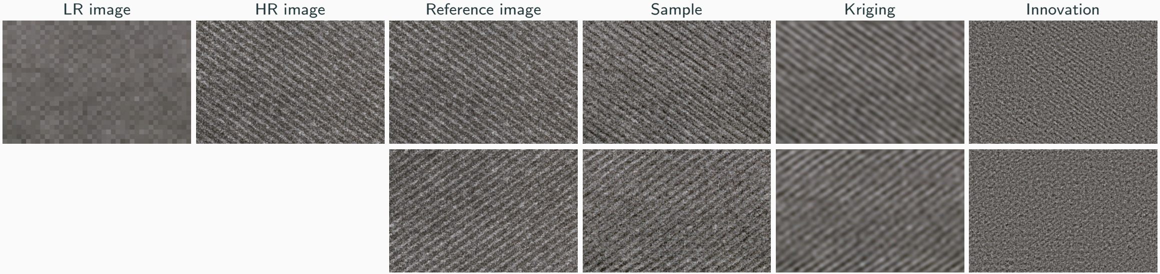
Let $\mathbf{U}_{\mathrm{HR}} \in \mathbb{R}^{\Omega_{M,N}}$ be a HR image, $\mathbf{U}_{\mathrm{LR}} = \mathbf{A}\mathbf{U}_{\mathrm{HR}}$ its LR version, $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ be the kriging operator and $\mathbf{X}_{\mathrm{SR}}$ the random image following the distribution of the SR samples then

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

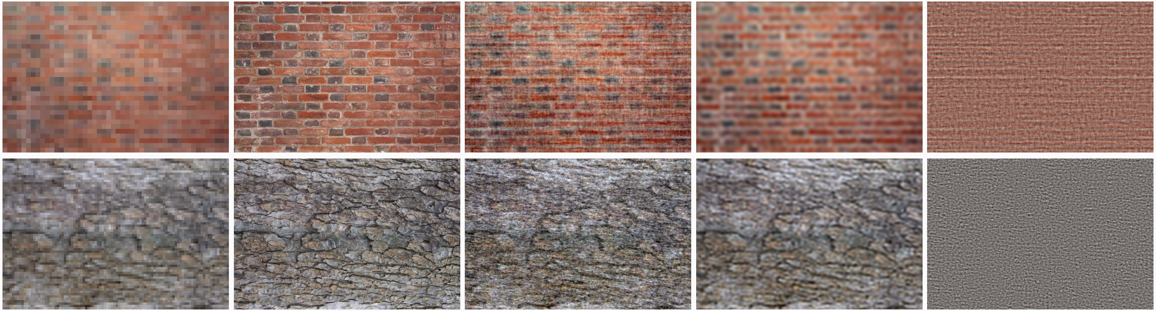
Simply put, the expected mean square error between the optimal HR image and Gaussian SR samples is always higher than the MSE between $\mathbf{U}_{\mathrm{HR}}$ and the associated kriging component $\mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}}$.

Fails of the method

Reference choice



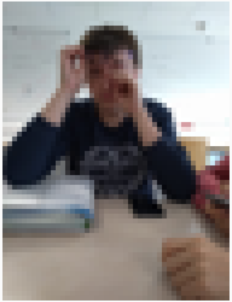
LR image	HR image	SR Gaussian sample	Kriging comp.	Innovation comp.
$\mathbf{U}_{\mathrm{LR}}$	$\mathbf{U}_{\mathrm{HR}}$	$\mathbf{U}_{\mathrm{SR}}$	$\mathbf{\Lambda}^T \mathbf{U}_{\mathrm{LR}}$	$\tilde{\mathbf{U}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{U}}$



→ Not stationary textures

Structured textures

LR image
$\mathbf{U}_{\mathrm{LR}}$

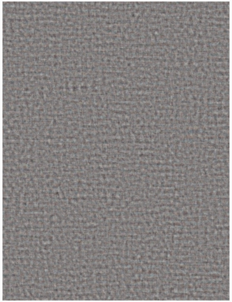


HR image
$\mathbf{U}_{\mathrm{HR}}$

SR Gaussian sample
$\mathbf{U}_{\mathrm{SR}}$

Kriging comp.
$\mathbf{\Lambda}^T\mathbf{U}_{\mathrm{LR}}$

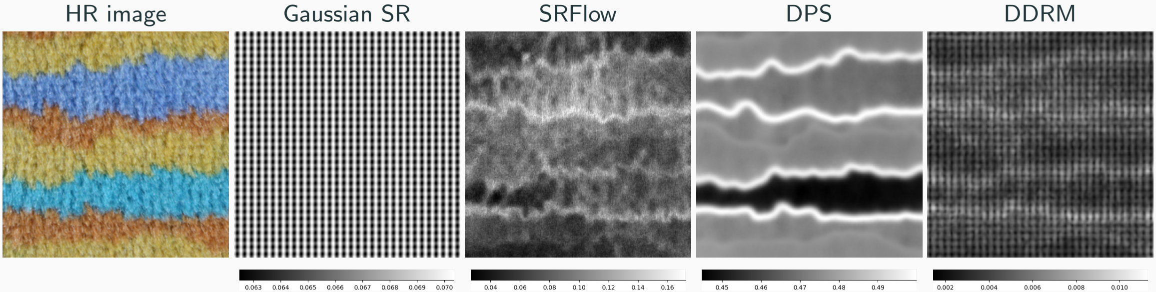
Innovation comp.
$\tilde{\mathbf{U}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{U}}$



HR image
$\mathbf{U}_{\mathrm{HR}}$

Kriging comp.
$\mathbf{\Lambda}^T \mathbf{U}_{\mathrm{LR}}$

Variance


HR image
Gaussian SR
SRFlow
DPS
DDRM
0.063 0.064 0.065 0.066 0.067 0.068 0.069 0.070
0.04 0.06 0.08 0.10 0.12 0.14 0.16
0.45 0.46 0.47 0.48 0.49
0.002 0.004 0.006 0.008 0.010

A little lie

The RGB convolution

An grayscale convolution acts like this:

$$\mathbf{\Gamma U} = \mathbf{t} \star \mathbf{U} \tag{3}$$

In Fourier,

$$\widehat{\mathbf{\Gamma U}} = \hat{\mathbf{t}} \odot \hat{\mathbf{U}} \tag{4}$$

An grayscale convolution acts like this:

$$\mathbf{\Gamma U} = \mathbf{t} \star \mathbf{U} \tag{3}$$

In Fourier,

$$\widehat{\mathbf{\Gamma U}} = \hat{\mathbf{t}} \odot \hat{\mathbf{U}} \tag{4}$$

An RGB convolution acts like this:

$$\left(\mathbf{\Gamma U}\right)_i = \mathbf{t}_i \star \check{\mathbf{t}_1} \star \mathbf{U}_1 + \mathbf{t}_i \star \check{\mathbf{t}_2} \star \mathbf{U}_2 + \mathbf{t}_i \star \check{\mathbf{t}_3} \star \mathbf{U}_3, \quad 1 \leqslant i \leqslant 3. \tag{5}$$

In Fourier,

$$\widehat{\left(\mathbf{\Gamma U}\right)}_i = \hat{\mathbf{t}}_i \odot \overline{\widehat{\mathbf{t}_1}} \odot \widehat{\mathbf{U}_1} + \hat{\mathbf{t}}_i \overline{\widehat{\mathbf{t}_2}} \odot \widehat{\mathbf{U}_2} + \hat{\mathbf{t}}_i \odot \overline{\widehat{\mathbf{t}_3}} \odot \widehat{\mathbf{U}_3}, \quad 1 \leqslant i \leqslant 3. \tag{6}$$

$$(\mathbf{\Gamma U})_i = \mathbf{t}_i \star \check{\mathbf{t}_1} \star \mathbf{U}_1 + \mathbf{t}_i \star \check{\mathbf{t}_2} \star \mathbf{U}_2 + \mathbf{t}_i \star \check{\mathbf{t}_3} \star \mathbf{U}_3, \quad 1 \leqslant i \leqslant 3. \tag{7}$$

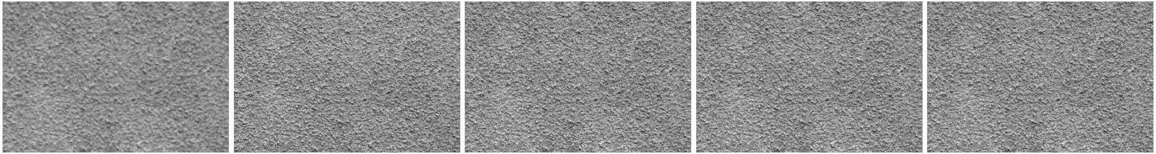
is replaced by

$$\left(\tilde{\mathbf{\Gamma}}\mathbf{U}\right)_i = \mathbf{t}_i \star \check{\mathbf{t}_i} \star \mathbf{U}_i, \quad 1 \leqslant i \leqslant 3. \tag{8}$$

Effect of RGB approximation

Grayscale example

LR image	HR image	Gaussian SR (ours)	CGD (10^2 steps)	CGD (10^6 steps)



RGB example

LR image	HR image	Gaussian SR (ours)	CGD (10^2 steps)	CGD (10^6 steps)

Effect of RGB approximation

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	$1.32\text{E}{-16}$	0.01	151.17	$2.54\text{E}{-1}$	0.01	37.94
CGD (10^2 steps)	$2.74\text{E}{-2}$	0.13	129.16	$2.73\text{E}0$	0.38	25.08
CGD (10^3 steps)	$1.03\text{E}{-3}$	0.76	147.49	$3.78\text{E}{-1}$	2.49	30.19
CGD (10^4 steps)	$4.72\text{E}{-5}$	7.44	167.60	$1.36\text{E}{-1}$	23.1	35.07
CGD (10^5 steps)	$1.30\text{E}{-8}$	81.3	145.31	$2.37\text{E}{-2}$	258	39.71
CGD (10^6 steps)	$2.69\text{E}{-42}$	749	-	$4.97\text{E}{-3}$	2588	-

Conclusion

Conclusion

- Our method has a limited scope but has a well-posed mathematical assumption.
- The stationarity assumption is very strong: details are affected in the SR samples.

What you missed :

- The pixel-wise variance can be computed.
- Study of stability.
- Can be used for other ill-posed problems with $\mathbf{A} = SC$.

Perspectives:

- Application to diffusion models.

Thank you for your attention !