Texture Inpainting Using Efficient Gaussian Conditional Simulation

Bruno Galerne, Arthur Leclaire

TEXTURE INPAINTING USING EFFICIENT GAUSSIAN CONDITIONAL SIMULATION

BRUNO GALERNE* AND ARTHUR LECLAIRE†

Abstract. Inpainting consists in computing a plausible completion of missing parts of an image given the available content. In the restricted framework of texture images, the image can be seen as a realization of a random field model, which gives a stochastic formulation of image inpainting: on the masked exemplar one estimates a random texture model which can then be conditionally sampled in order to fill the hole.

In this paper is proposed an instance of such stochastic inpainting methods, dealing with the case of Gaussian textures. First a simple procedure is proposed for estimating a Gaussian texture model based on a masked exemplar, which, although quite naive, gives sufficient results for our inpainting purpose. Next, the conditional sampling step is solved with the traditional algorithm for Gaussian conditional simulation. The main difficulty of this step is to solve a very large linear system, which, in the case of stationary Gaussian textures, can be done efficiently with a conjugate gradient descent (using a Fourier representation of the covariance operator). Several experiments show that the corresponding inpainting algorithm is able to inpaint large holes (of any shape) in a texture, with a reasonable computational time. Moreover, several comparisons illustrate that the proposed approach performs better on texture images than state-of-the-art inpainting methods.

Key words. Inpainting, Gaussian textures, Conditional simulation, Simple kriging.

AMS subject classifications. 62M40, 68U10, 60G15

1. Introduction.

Inpainting consists in filling missing or corrupted regions in images by inferring from the context. In other words, given an image whose pixel values are missing in a masked domain, the problem is to propose a possible completion of the mask that will appear as natural as possible given the available part of the image. Inspired by art restorers, this problem was called "inpainting" by Bertalmio et al. [8], but was already addressed under the name "disocclusion" in [68, 67]. Both these works suggest to fill the hole by extending the geometric structures, either by level-lines completion [68] or by iterating a finite-difference scheme [8]. These early methods already give good results on structured images provided that the mask is sufficiently thin. However, they fail to inpaint textural content, which is the main purpose of this paper.

General image inpainting is a very ill-posed problem, and instead of retrieving the occluded content, one can only make a guess of what the image should have been. However, in the restricted framework of textures, we have at our disposal several stochastic models which can be used to model and synthesize a large class of textures. In this setting, inpainting consists in first estimating a stochastic model from the unmasked region, and then performing conditional simulation of the estimated random model given the values around the mask. This point of view thus provides a better-posed formulation of textural inpainting, which has been seldom considered in the past. In particular, such approximate conditional sampling results are given in [31, 84, 54] under the name "constrained texture synthesis". Also, the authors of [24] give an instructive discussion which opposes deterministic and stochastic strategies for image inpainting (with the intention to explain the differences between [31] and [84]).

It seems reasonable to assert that the choice between deterministic methods or stochastic methods must be driven by the level of randomness of the data. Here, we

*Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité, France.
†CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France.

will mainly focus on inpainting very irregular texture images, called microtextures. Following the definition of [36], microtextures are images whose visual perception is not affected by randomization of the Fourier phase. These textures are not well described by a generic variational principle. In contrast, they can be precisely and efficiently synthesized with simple stochastic models that rely on second-order statistics, for example the asymptotic discrete spot noise (ADSN) introduced in [83] and thoroughly studied in [36, 86, 59]. In this paper, we propose a microtexture inpainting algorithm that relies on a precise conditional sampling. Conditional sampling of the ADSN model can be easily formulated, and gives inpainting results which are visually better than the ones obtained with recent methods while keeping strong mathematical guarantees.

In the remaining paragraphs of this introduction, we discuss existing inpainting techniques, and in particular discuss the links between image inpainting and texture synthesis. Giving an exhaustive overview of the literature on this famous problem is not the main purpose of this paper. We refer the interested reader to [44, 15, 78] for much more detailed reviews of existing methods.

1.1. Inpainting Algorithms for Geometric Content. As mentioned above, a very natural way to inpaint images is to propagate the geometric content through the masked region. To that purpose, the early geometric inpainting methods described by Masnou and Morel [68, 67] consist in connecting the level lines across the hole in order to satisfy the Gestaltist's principle of good continuation. More precisely, the inpainted image is the solution of a generic minimization problem which includes the total variation (TV) of the image and the angle total variation of the level lines (Euler's elastica).

Closely related to these generic variational inpainting methods lie models based on partial differential equations (PDE). Bertalmio et al. [8] suggest to iterate a finite-difference scheme, which was later interpreted as a numerical scheme for a PDE related to Navier-Stokes equation [7]. Of course, there is a strong connection between PDE-based and variational methods because the minimum of a generic functional satisfies the associated Euler-Lagrange equation (but a PDE may not be associated with a variational problem [78]). Among many papers lying in between PDEs and generic variational problems, we will only quote a few important contributions.

Ballester et al. [5] propose to perform joint interpolation of image values and gradient orientations by solving a minimization problem which leads to coupled second-order PDEs on image values and gradient orientations. Chan and Shen [18] give a detailed study of the inpainting method based on TV minimization (which, compared to [68] drops the elastica term in the minimization problem), and propose a more general scheme called curvature-driven diffusion (which allows to better respect the good continuation principle). The link with Mumford-Shah image model was already discussed in [18], and more importantly exploited by Esedoglu and Shen [34], who completed the Mumford-Shah model with an Euler's elastica term, leading to fourth-order nonlinear parabolic PDEs, and allowing better connectivity in the inpainting result. Later, other fourth-order PDEs were exploited to inpaint non-texture images with better connectivity: Bertozzi et al. [10] propose to solve a modified Cahn-Hilliard equation for fast inpainting of binary or highly-contrasted images, an approach which was generalized to real-valued images by Burger et al. [14]. Finally, Bornemann and März [12] propose an efficient non-iterative inpainting algorithm which is based on a transport equation and inspired by the fast marching algorithm of [81].

A common drawback of these deterministic methods is that they are not able

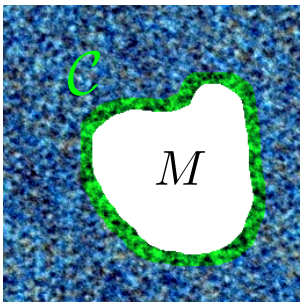

FIG. 1. ***Textural inpainting via conditional simulation.*** *Inpainting with a stochastic texture model amounts to sampling the values on the mask M knowing the values on conditioning points $\mathcal{C}$ located at the border of the mask.*

to inpaint textural content precisely because solving a PDE or a variational problem often imposes a certain degree of smoothness for the solution.

1.2. Exemplar-based Inpainting, Sampling or Minimizing?... An efficient way to model irregular images is to consider stochastic image models, and in particular many texture synthesis algorithms can be formulated as sampling a probability distribution. Thus, one first strategy to inpaint textural parts of an image is to use an exemplar-based texture synthesis algorithm and to blend the synthesized content in the masked image. Such a method was proposed by Igehy and Pereira [49] who relied on Heeger-Bergen synthesis algorithm [47] to produce textural content.

On the other hand, if a stochastic image model is fixed, inpainting can be understood as sampling a conditional distribution, as illustrated on Fig. 1. This point of view was originally adopted by Efros and Leung [31]. These authors suggest to approximate conditional sampling of a Markov random field (MRF) model by progressive completion of the unknown region using patch nearest neighbor search. Even if they show some texture inpainting results, their main concern is structured texture synthesis. For inpainting, this patch-based approach was precised in [11, 24]. In particular, Demanet et al. discuss the two possible formulations of the inpainting problem as either minimizing the energy E or sampling the probability distribution Ce^{-E}. They give several arguments to support that the variational point of view is a lighter and sufficient method to efficiently compute an inpainting solution. However, let us mention that the patch-based energy given in [24] is highly non-convex, and that the adopted optimization strategy does not offer much theoretical guarantees. Therefore, the empirical conclusions based on the results of this algorithm must be interpreted carefully. Our paper will shed some more light on this interesting (and still open) question, in the case of Gaussian textures.

Many other inpainting methods were inspired by these exemplar-based synthesis algorithms [24, 30, 23, 71, 53, 85, 4, 13, 87, 3, 2, 57, 63, 46, 69, 15]. These papers contain several clever algorithmic extensions of the original algorithm of [24]. In particular, Criminisi et al. [23] highlighted the importance of the pixel-filling order, and suggested that it should be driven by (progressively updated) patch priorities measuring the amount of available data and the quantity of structural information in the currently synthesized content. Many authors [30, 53, 85, 3, 69] demonstrated that the inpainting problem could be more efficiently solved (both in visual terms or numerical terms) by relying on a multi-scale strategy. From a computational point of view, the speed of these algorithms highly depends on the method used for getting

patch nearest neighbors, and many state of the art methods rely on the PatchMatch method which efficiently computes an approximate nearest neighbor field [6, 2, 63, 69]. Let us also mention that the choice of the metric used for patch comparison may influence the inpainting results; to that purpose, the authors of [63, 69] suggested to improve the comparison by including textural features in the patch distance (e.g. local sum of absolute derivatives).

Here we would like to put the emphasis on a few papers which provide a thorough mathematical analysis of the variational formulation proposed by [24]. Aujol et al. [4] show the existence of a solution to a continuous analog of Demanet et al.' energy among the set of piecewise roto-translations, propose several extensions of this problem (allowing for either regularization or cartoon+texture decomposition), and also provide a 2D-example which illustrates the model ability to globally reconstruct geometric features. Arias et al. [3] propose and compare several variational models obtained by varying the distance used in patch comparison (using the L^1 or L^2 norm on the image values or gradients), and also propose to replace the patch correspondence by generalized patch linear combinations using an adaptive weighting function. In [2], the same authors provide an additional mathematical analysis with a proof of the solution existence, of the convergence of the proposed minimization algorithm. In these works, the inpainting problem is mainly formulated with a correspondence map (or a more general weighting function in [3]). In contrast, Liu and Caselles have shown in [63] that using an offset map instead allows to formulate inpainting as a discrete optimization problem which is efficiently solved with graph cuts. The statistics of patch offsets have been studied in [46]; He and Sun compute and exploit recurrent patch offsets in order to simplify the graphcut inpainting approach leading to an even faster algorithm.

Finally, the above-mentioned structural and exemplar-based methods can be combined to obtain hybrid structure-texture inpainting methods [9, 50, 80, 17]. Also, several authors proposed inpainting methods based on sparse decompositions of images or patches [32, 64, 16, 72]. In these methods, the inpainting is also formulated as a minimization problem (which can be coupled with the dictionary learning problem as in [64]). Although these methods are efficient in recovering missing data for thin or randomly-distributed masks, they are not able to fill large missing regions.

1.3. Gaussian Conditional Simulation.

In this paper, we will address textural inpainting by precise conditional sampling of a stochastic texture model.

In the computer graphics community, many authors have demonstrated the expressive power of microtexture models based on Fourier phase randomization [60, 61] or on convolution of spot functions with noisy patterns [83]. Later, these models were studied in more detail by Galerne et al. [36] who propose in particular a simple analysis-synthesis pipeline for by-example microtexture synthesis with the Asymptotic Discrete Spot Noise (ADSN) model (which is the Gaussian limit of Van Wijk's Spot Noise model [83]). Such a Gaussian model is described by its first and second-order moments, and allows for fruitful mathematical developments, with applications in texture analysis [25], texture mixing [86], procedural texture synthesis [38, 40].

In this paper (following the preliminary work of [39]), we propose to take advantage of another benefit of the Gaussian model, which is the availability of a precise conditional sampling algorithm. Indeed, for Gaussian vectors, independence is equivalent to uncorrelatedness, which can be rephrased as orthogonality in the Hilbert space of square-integrable random variables. Therefore, conditional simulation of a zero-mean Gaussian vector F only requires to compute an orthogonal projection F^*

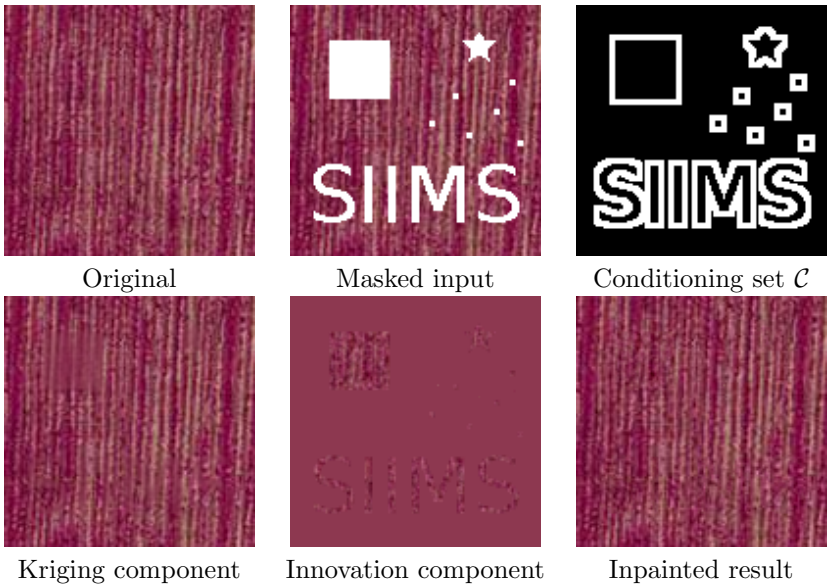

FIG. 2. ***Summary of our microtexture inpainting method*** *The main idea of our method is to fill the masked region with a conditional sample of a Gaussian model. So this method is less about retrieving the initial image than computing another plausible sample of the texture model in the masked region. The Gaussian model is estimated from the unmasked values, and conditionally sampled knowing the values on a set $\mathcal{C}$ composed of a 3 pixel wide border of the mask. The conditional sample is obtained by adding a kriging component (derived from the conditioning values) and an innovation component (derived from an independent realization of the Gaussian model). The former extends the long-range correlations and the latter adds texture details, in a way that globally preserves the global covariance of the model. Though limited to microtextures, this algorithm is able to fill both small and large holes, whatever the regularity of the boundary.*

on a subspace of random variables (which corresponds to the conditional expectation given the known values) and to sample the orthogonal component $F - F^*$. Following the presentation of [55], we will rely on the terminology which is traditionally used in "simple kriging estimation": the conditional expectation F^* will be called "kriging component", and $F - F^*$ will be called "innovation component". The role of these two components for conditional simulation is illustrated in Fig. 2. Let us mention that in the Gaussian case, solving the maximum *a posteriori* for the conditional model amounts to computing the conditional expectation (i.e. kriging component), which is very different from conditional sampling, as one can see on Fig. 2.

To the best of our knowledge, microtexture inpainting has not been addressed in those terms in the past. Gaussian conditional simulation algorithm was used by Hoffman and Ribak [48] for cosmological constrained simulations with parametric Gaussian models. More recently, local Gaussian conditional models were used for structured texture synthesis in [75, 74]. In the monoscale version [75], Raad et al. suggest to progressively sample the texture with conditional sampling of local Gaussian models estimated from the exemplar (with nearest neighbor search as in [31, 84]); they also propose a multiscale adaptation of this algorithm [74]. As for [31], this algorithm could also be adapted for inpainting, but, because of the progressively estimated local models, the global model is not Gaussian. Ordinary kriging was used by Chandra et al. [19] to interpolate sparsely sampled textural data (but does not compute a conditional sample).

1.4. Connections with geostatistics. However, kriging-based Gaussian conditional simulation is a traditional method used for data interpolation in geostatistics [21, 55, 22, 43, 26]. Several parts of the method we propose are already well-known to geostatisticians, sometimes under other names. In particular, the ADSN model that we use is an instance of moving-average random fields [51, 70] whose spectral-based unconditional sampling algorithm is explained in [45, 20, 58]. The authors of [58] also suggest an optimization procedure to modify the unconditional sample so that it complies with the available data. In contrast, we propose direct sampling of a global conditional Gaussian model. Let us emphasize that, contrary to many examples shown in the geostatistics literature, our imaging application leads to very large conditioning sets (with possibly several thousands conditioning values). Thus, in our case, precise conditional sampling is much more difficult than unconditional sampling.

Also, in the geostatistics literature, several authors have proposed generalized kriging algorithms for data prediction with various stochastic models [76, 1, 77, 33, 21, 62]. In particular, in [76], Rue proposes a fast algorithm for conditional simulation in the particular case of Gaussian Markov random fields. Another technique for fast sampling in geostatistics is given by sequential simulation [42], which amounts to progressive filling of the pixels in a random order using successive conditional sampling. In our context, this approach would require to solve larger and larger kriging systems and would not be as efficient as our global approach. About progressive filling of the pixels, let us also mention a clear connection between the inpainting adaptation of [31] and the direct sampling method of [66]. We refer the interested reader to [65] for a much deeper discussion on the links between texture synthesis and multiple-point geostatistics.

1.5. Plan of the Paper. In Section 2, we explain the traditional algorithm for Gaussian conditional simulation (using a terminology that is derived from kriging estimation). In Section 3, we apply this conditional sampling algorithm to microtexture inpainting. In particular, we discuss the estimation of a Gaussian model on a masked exemplar, and we also provide a Fourier based algorithm which allows to compute the kriging estimation even when the number of conditioning points is very large. Finally, in Section 4, we provide several texture inpainting experiments to illustrate the validity of our approach; in particular we show that our method can compete with state of the art inpainting methods on textural content.

2. Gaussian Conditional Simulation. In this section, we recall the classical algorithm for conditional sampling of Gaussian random vectors. Following [55], we rely on a kriging framework that we introduce next.

Notation. Let Ω be a finite set. Let $(F(x))_{x\in\Omega}$ be a real-valued Gaussian vector, that is, a real-valued random vector for which any linear combination of the components is Gaussian. We assume that F has **zero mean**. The covariance of F is written $\Gamma(x,y) = \mathrm{Cov}(F(x),F(y)) = \mathbb{E}(F(x)F(y)),\ x,y\in\Omega$. For a set $A\subset\Omega$ and a function $f:\Omega\to\mathbb{R}$ we denote by $|A|$ the cardinality of the finite set A, and $f_{|A}$ the restriction to A of the function f.

We also introduce a subset $\mathcal{C}\subset\Omega$ of conditioning points. Given prescribed values $\varphi:\mathcal{C}\to\mathbb{R}$ on $\mathcal{C}$, conditional Gaussian simulation consists in sampling the conditional distribution of F given that $F_{|\mathcal{C}} = \varphi$. As we shall see later, this conditional sampling makes sense as soon as φ belongs to the support of the distribution of $F_{|\mathcal{C}}$, which is the range of the restricted covariance matrix $\Gamma_{|\mathcal{C}\times\mathcal{C}}$ and denoted by $\mathrm{Range}(\Gamma_{|\mathcal{C}\times\mathcal{C}})$.

2.1. Simple Kriging Estimation. We define the **simple kriging estimator**

$$F^*(x) = \mathbb{E}(\ F(x) \mid F(c)\ ,\ c \in \mathcal{C}\ ). \tag{1}$$

A standard result of probability theory [28] ensures that in the Gaussian case $F^*(x)$ is the orthogonal projection of $F(x)$ on the subspace of linear combinations of $(F(c))_{c\in\mathcal{C}}$ (for the L^2-distance between square-integrable random variables). Hence, there exist deterministic coefficients $(\lambda_c(x))_{c\in\mathcal{C}}$, called **kriging coefficients** such that

$$F^*(x) = \sum_{c\in\mathcal{C}} \lambda_c(x) F(c). \tag{2}$$

Notice that by definition, $F^*(x) = F(x)$ for every $x \in \mathcal{C}$.

Generally speaking, for a given x, there may be several possible sets of kriging coefficients i.e. several vectors $(\lambda_c(x))_{c\in\mathcal{C}}$ which satisfy (2) (for example if there are two distinct points $c_1, c_2 \in \mathcal{C}$ such that $F(c_1) = F(c_2)$). But we will later give a canonical way to compute a valid set of kriging coefficients.

2.2. Gaussian Conditional Sampling Using Kriging Estimation. Let us fix a set of coefficients $(\lambda_c(x))_{x\in\Omega,c\in\mathcal{C}}$ satisfying (2). For any $\varphi : \mathcal{C} \to \mathbb{R}$, we denote by φ^* the kriging estimation based on the values φ, defined for $x \in \Omega$ by $\varphi^*(x) = \sum_{c\in\mathcal{C}} \lambda_c(x)\varphi(c)$. With a notation abuse, if $\varphi : \Omega \to \mathbb{R}$, we will denote $\varphi^* = (\varphi_{|\mathcal{C}})^*$.

THEOREM 1 (See for example [28, 55]). *F^* and $F - F^*$ are independent. Consequently, if G is independent of F with same distribution, then $H = F^* + (G - G^*)$ has the same distribution as F and satisfies $H_{|\mathcal{C}} = F_{|\mathcal{C}}$.*

If $\varphi_{|\mathcal{C}} \in \mathrm{Range}(\Gamma_{|\mathcal{C}\times\mathcal{C}})$, a conditional sample of F given $F_{|\mathcal{C}} = \varphi_{|\mathcal{C}}$ can thus be obtained with $\varphi^* + F - F^*$. In this decomposition, φ^* will be called the **kriging component** and $F - F^*$ will be called the **innovation component**.

2.3. Expression of the Kriging Coefficients. In order to compute the kriging estimator at $x \in \Omega$, one needs to compute a valid set of kriging coefficients $(\lambda_c(x))_{c\in\mathcal{C}}$. Since F^* and $F - F^*$ are orthogonal, we get that the row vector $\lambda(x) = (\lambda_c(x))_{c\in\mathcal{C}}$ is a solution of the following $|\mathcal{C}| \times |\mathcal{C}|$ linear system

$$\forall c \in \mathcal{C}, \quad \sum_{d\in\mathcal{C}} \lambda_d(x)\Gamma(d,c) = \Gamma(x,c), \quad \text{i.e.} \quad \lambda(x)\Gamma_{|\mathcal{C}\times\mathcal{C}} = \Gamma_{|\{x\}\times\mathcal{C}}. \tag{3}$$

Conversely, any solution of (3) gives a valid set of kriging coefficients satisfying (2).

Aggregating the kriging coefficients in a $|\Omega| \times |\mathcal{C}|$ matrix $\Lambda = (\lambda_c(x))_{x\in\Omega,c\in\mathcal{C}}$, the system characterizing the kriging coefficients can also be written $\Lambda\Gamma_{|\mathcal{C}\times\mathcal{C}} = \Gamma_{|\Omega\times\mathcal{C}}$. If the matrix $\Gamma_{|\mathcal{C}\times\mathcal{C}}$ is invertible, the global system admits a unique solution $\Lambda = \Gamma_{|\Omega\times\mathcal{C}}\Gamma^{-1}_{|\mathcal{C}\times\mathcal{C}}$. In the case where $\Gamma_{|\mathcal{C}\times\mathcal{C}}$ is not invertible, it is always possible to compute valid kriging coefficients with the pseudo-inverse $\Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}$. Indeed, since the system (3) has a solution[1], then $\Gamma_{|\{x\}\times\mathcal{C}}\Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}$ is also a solution. Thus we can always consider the set of kriging coefficients given by $\Lambda = \Gamma_{|\Omega\times\mathcal{C}}\Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}$.

Once a set Λ of valid kriging coefficients has been computed, a conditional sample of F given $F_{|\mathcal{C}} = \varphi$ can be obtained as $\Lambda\varphi + F - \Lambda F_{|\mathcal{C}}$, where φ and F are written as column vectors.

[1] The existence of such a solution directly comes from the existence of the orthogonal projection of $F(x)$ on the subspace spanned by the $F(c), c \in \mathcal{C}$.

2.4. Matrix Expression of the Conditional Simulation. From this expression of the conditional sample, we will derive the usual expression of the Gaussian conditional distribution in matrix notation (as e.g. in [77, 74]).

Let $p = |\mathcal{C}|$, $q = |\Omega \setminus \mathcal{C}|$ (where $\Omega \setminus \mathcal{C}$ denotes the complement of $\mathcal{C}$ in Ω) and $n = |\Omega|$. Let us introduce the matrices $R = \begin{pmatrix} I_p & 0 \end{pmatrix} \in \mathbb{R}^{p \times n}$, $S = \begin{pmatrix} 0 & I_q \end{pmatrix} \in \mathbb{R}^{q \times n}$, Using the first p indices for the elements of $\mathcal{C}$, we write block decompositions

$$F = \begin{pmatrix} F_{|\mathcal{C}} \\ F_{|\Omega\setminus\mathcal{C}} \end{pmatrix} = \begin{pmatrix} RF \\ SF \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \Gamma_{|\mathcal{C}\times\mathcal{C}} & \Gamma_{|\mathcal{C}\times(\Omega\setminus\mathcal{C})} \\ \Gamma_{|(\Omega\setminus\mathcal{C})\times\mathcal{C}} & \Gamma_{|(\Omega\setminus\mathcal{C})\times(\Omega\setminus\mathcal{C})} \end{pmatrix} = \begin{pmatrix} R\Gamma R^T & R\Gamma S^T \\ S\Gamma R^T & S\Gamma S^T \end{pmatrix}.$$

With such notation, if $\varphi \in \mathrm{Range}(\Gamma_{|\mathcal{C}\times\mathcal{C}})$, a conditional sample of F given $F_{|\mathcal{C}} = \varphi$ is given by $\Lambda\varphi + F - \Lambda RF$. From this expression we get the conditional distribution

$$F \mid F_{|\mathcal{C}} = \varphi \quad \sim \quad \mathcal{N}\big(\ \Lambda\varphi\ ,\ (I_n - \Lambda R)\Gamma(I_n - \Lambda R)^T\ \big). \tag{4}$$

Using the kriging system (which rewrites $\Lambda R\Gamma R^T = \Gamma R^T$), we get the usual formulae

$$\mathbb{E}(\ SF \mid F_{|\mathcal{C}} = \varphi\ ) = S\Lambda\varphi = S\begin{pmatrix} R\Gamma R^T \\ S\Gamma R^T \end{pmatrix}(R\Gamma R^T)^\dagger\varphi = S\Gamma R^T(R\Gamma R^T)^\dagger\varphi, \tag{5}$$

$$\mathrm{Cov}(\ SF \mid F_{|\mathcal{C}} = \varphi\ ) = S\Gamma S^T - S\Gamma R^T(R\Gamma R^T)^\dagger R\Gamma S^T. \tag{6}$$

When $R\Gamma R^T = \Gamma_{|\mathcal{C}\times\mathcal{C}}$ is non-singular, we get back the expressions of [77, 74].

3. Microtexture Inpainting Algorithm. This section contains our main contribution: how to use Gaussian conditional sampling for microtexture inpainting.

We are given an input texture image $u : \Omega \to \mathbb{R}$ defined on a finite rectangular domain $\Omega \subset \mathbb{Z}^2$. The values of u are known except on the mask $M \subset \Omega$ and we want to generate plausible values on the mask given the surrounding content. For that, we sample a stationary Gaussian texture model $(U(x))_{x\in\Omega}$ given the values of u outside M. More precisely, we consider a Gaussian model associated with an asymptotic discrete spot noise (ADSN), which we sample knowing the values on a conditioning set $\mathcal{C} = \partial_w M$ defined as the outer border of M with width w pixels (we usually take $w = 3$ but we discuss this choice in Section 4.4).

After recalling the basics about the ADSN model, we discuss the estimation of such a model on a masked exemplar texture. Then we give an efficient and scalable way to compute the kriging estimator for the ADSN model by relying on conjugate gradient descent (numerical issues are discussed in the IPOL companion paper [37]). Visual results are given in the next section.

3.1. ADSN Models. As shown in [83, 36], a convenient model for microtexture is given by the asymptotic discrete spot noise (ADSN). Given a function $h : \mathbb{Z}^2 \to \mathbb{R}$ with finite support, the ADSN corresponding to h is the convolution of h with a normalized Gaussian white noise W on $\mathbb{Z}^2$, defined as

$$\forall x \in \mathbb{Z}^2, \quad h * W(x) = \sum_{y\in\mathbb{Z}^2} h(y)W(x-y). \tag{7}$$

This Gaussian random field is stationary, has zero mean, and its covariance function is given by $\mathbb{E}(h*W(x)h*W(y)) = (h*\tilde{h})(x-y)$, where $\tilde{h}(z) = h(-z)$. The restriction on a finite $\Omega \subset \mathbb{Z}^2$ of $h*W$ is a zero-mean Gaussian model $(F(x))_{x\in\Omega}$. Thanks to the simple convolutive expression of the ADSN, it can be efficiently sampled using the fast Fourier transform (FFT). Depending on the boundary conditions, we can consider a

periodic ADSN or a non-periodic ADSN. Apart from a slight gain of complexity, there is no general reason to favor the periodic model. The choice is often driven by the applicative context; for example, non-periodic models are better suited for on-demand texture synthesis [38, 40]. Here we choose the non-periodic model and we refer to [59, Chap.2] for a detailed exposure regarding both ADSN models.

Extension to Color Images. ADSN models extend to color images by convolving each color channel with the same white noise in (7). This gives an $\mathbb{R}^d$-valued Gaussian random field F on Ω (where d is the number of channels, i.e. 3 for color images). Regarding the conditional simulation, a simple way to understand this extension is to consider the $\mathbb{R}^d$-valued random field F as a real-valued random field on $\Omega \times \{1, \dots, d\}$. The covariance matrix is then given by

$$\forall (x,j),(y,k) \in \Omega \times \{1,\dots,d\}, \quad \Gamma((x,j),(y,k)) = \mathbb{E}(F_j(x)F_k(y)). \tag{8}$$

Even if this changes the covariance matrix, we keep the same notation for restrictions of the covariance matrix: for example, we still use the notation $\Gamma_{|\mathcal{C}\times\mathcal{C}}$ for the covariance of F on $\mathcal{C}$, but strictly speaking we should write $\Gamma_{|(\mathcal{C}\times\{1,\dots,d\})\times(\mathcal{C}\times\{1,\dots,d\})}$.

3.2. Estimation of the Gaussian Model. If the image $u : \Omega \to \mathbb{R}^d$ were entirely available, the estimation procedure would be the same as for texture synthesis [36, 38], which is briefly recalled here. We compute the mean value $\bar{u} = \frac{1}{|\Omega|}\sum_{x\in\Omega} u(x)$ and the normalized spot $t_u = \frac{1}{\sqrt{|\Omega|}}(u - \bar{u})$ (extended by zero-padding). The microtexture u is then synthesized by sampling $\bar{u} + t_u * W$, with W a normalized Gaussian white noise. We call *oracle model* the ADSN model estimated from the unmasked exemplar.

In the inpainting context, only the values on $\Omega \setminus M$ are available. Thus, we choose a subdomain $\omega \subset \Omega \setminus M$ and we derive an ADSN model using the restriction $v = u_{|\omega}$. A simple way to do that is to consider the Gaussian model $U = \bar{v} + t_v * W$ where

$$\bar{v} = \frac{1}{|\omega|}\sum_{x\in\omega} v(x), \quad t_v(x) = \begin{cases} \frac{1}{\sqrt{|\omega|}}(v(x) - \bar{v}) & \text{if } x \in \omega, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

This choice amounts to estimate the texture covariance by $c_v = t_v * \tilde{t}_v^{\,T}$, which writes

$$c_v(h) = \frac{1}{|\omega|}\sum_{x\in\omega\cap(\omega-h)} (u(x+h) - \bar{v})(u(x) - \bar{v})^T \in \mathbb{R}^{d\times d}. \tag{10}$$

This subdomain ω is not constrained to be a rectangle; for example, a canonical choice would be to consider $\omega = \Omega \setminus M$. As will be observed in Section 4.2, this choice already gives good results in our inpainting framework. However, one must be aware that the geometry of ω may impact the quality of the estimation. We illustrate this effect in Fig. 3. In general, we observed that the performance of the naive estimator is surprisingly good provided that the mask is not too much irregular.

We would like to point out here that designing more precise estimators of the covariance is an interesting question. In particular, at first sight one can be puzzled by the normalization of (10). A better normalized estimator $c'_v(h)$ would be obtained by replacing $\frac{1}{|\omega|}$ by $\frac{1}{|\omega\cap(\omega-h)|}$ in this formula. But a drawback of this new estimator is that it does not define a semi-definite positive estimator, and thus is not associated with a Gaussian model that could be sampled. A way to cope with this effect is to enforce semi-definite positiveness, which in the stationary case is equivalent to project

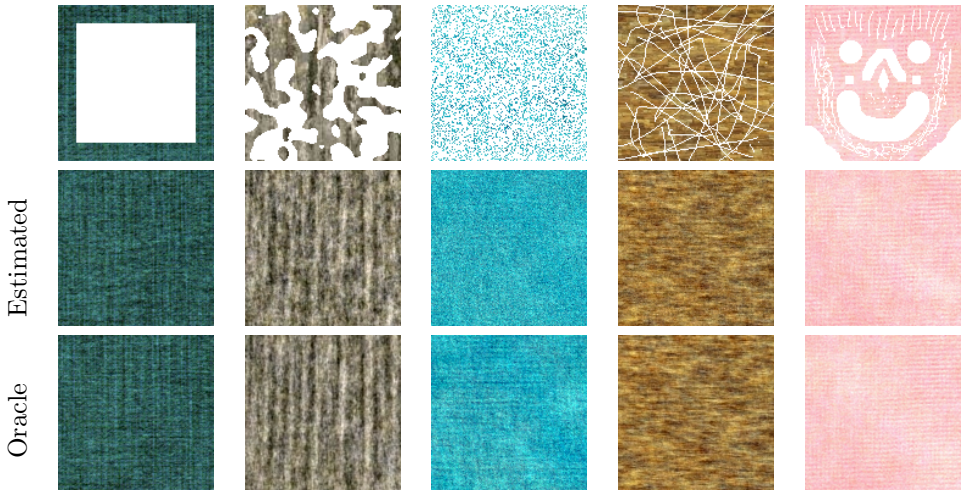

FIG. 3. ***Estimation of an ADSN model on a masked exemplar.*** *We illustrate with several types of mask the estimation of the Gaussian model with the naive estimator* (10) *using* $\omega = \Omega \setminus M$. *We display in the first row the masked exemplar, in the second row a sample of the estimated ADSN model, and in the third row a sample of the oracle ADSN model estimated from the unmasked exemplar (generated with the same random seed). As one can see, in terms of synthesis, the naive estimator produces nearly perfect results as soon as the mask complement contains a sufficiently large connected region to capture the textural aspect. The worst case is encountered for very irregular masks like the one shown in the third column (75% of masked pixels).*

on the non-negative orthant in Fourier domain. We have led some experiments in this direction, and they have shown that the resulting Gaussian model is not better than the one obtained with the naive estimator (both in terms of resynthesis or in terms of optimal transport distance between Gaussian models [38]). Indeed, the projection on the Fourier orthant has a dramatic impact on the model (in particular, it may significantly impact the estimation of the marginal variance).

One explanation of the success of the naive estimator for regular masks is that in this case we have $\frac{|\omega \cap (\omega - h)|}{|\omega|} \approx 1$ when $h \approx 0$. Therefore the naive estimator is approximately well normalized around 0 and thus correctly estimates the covariance in a neighborhood of 0, which is the most important part for microtexture images.

3.3. Kriging Estimation with Conjugate Gradient Descent. In this section, we propose an efficient way to compute a conditional sample of the ADSN model. The most difficult part consists in solving a large linear system involving the conditional values. This step is dealt with by using a conjugate gradient descent algorithm, which proves to be efficient even for very large images.

In order to draw a conditional sample on the mask M, we introduce a set of conditioning points $\mathcal{C} \subset \Omega \setminus M$. Ideally, we should choose $\mathcal{C} = \Omega \setminus M$; but we will see below that for computational and theoretical reasons, taking $\mathcal{C} = \partial_w M$ (border of M with width w) may be useful. Of course, in the case where $\mathcal{C} \subsetneq \Omega \setminus M$, we draw a conditional sample on Ω but we exploit only the restriction on M to get the inpainting result (in other words, on $\Omega \setminus M$ we always impose the original image).

As explained in the last section, after subtracting the estimated mean $\bar{v}$, we can use the ADSN model $(F(x))_{x \in \Omega}$ corresponding to the spot t_v (which is a zero mean Gaussian vector). Using the framework and notation of Section 2, we draw a

conditional sample $(F(x))_{x\in\Omega}$ given $F_{|\mathcal{C}} = u_{|\mathcal{C}} - \bar{v}$ by computing

$$(u-\bar{v})^* + F - F^* = \Lambda\big((u-\bar{v})_{|\mathcal{C}}\big) + F - \Lambda(F_{|\mathcal{C}}). \tag{11}$$

Let us explain how to efficiently apply the matrix $\Lambda = \Gamma_{|\Omega\times\mathcal{C}}\Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}$ to a given $\varphi \in \mathbb{R}^{\mathcal{C}}$.

Let us begin with the multiplication by $\Gamma_{\Omega\times\mathcal{C}}$, which is easier. Assume that $\psi = \Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}\varphi$ has been computed. Using the notation of Section 2.4, $\Gamma_{|\Omega\times\mathcal{C}}\psi = \Gamma\Psi$, where $\Psi = R^T\psi \in \mathbb{R}^{\Omega}$ is the zero-padding extension of ψ. Now, since Γ is the covariance function of an ADSN model, it can be simply computed by convolution. More precisely, $\Gamma\Psi$ is the restriction on Ω of the convolution of Ψ by $t_v * \tilde{t}_v$.

Computing $A^{\dagger}\varphi$ where $A = \Gamma_{|\mathcal{C}\times\mathcal{C}}$ is more costly. Assume for a moment that A is invertible. Then computing $A^{-1}\varphi$ amounts to solving a linear system of size $p\times p$ (where $p = d|\mathcal{C}|$). Since A is symmetric positive-definite, this can be reduced to solving two triangular systems thanks to the Cholesky factorization of A. Nevertheless, finding the Cholesky factorization of A requires $\mathcal{O}(p^3)$ flops in general. Therefore, this direct method will only work for small values of p. This was a major limitation of our preliminary work presented in [39].

To cope with this problem, we propose here to solve the linear system with a conjugate gradient descent algorithm, taking profit of the fact that applying the matrix A can be done efficiently. Indeed, computing $A\psi$ amounts to extend ψ to Ω by zero-padding, convolve by $t_v * \tilde{t}_v$ and restrict the result on $\mathcal{C}$. Besides, using a conjugate gradient descent on the normal equations allows to cope with possibly singular matrices A.

Following [52], we compute $A^{\dagger}\varphi$ by performing a conjugate gradient descent on

$$f : \psi \longmapsto \frac{1}{2}\|A\psi - \varphi\|^2 \tag{12}$$

with initialization $\psi_0 = 0$. This optimization procedure actually solves the normal equations $A^TA\psi = A^T\varphi$, which are equivalent to $A\psi = \varphi$ when $\varphi \in \mathrm{Range}(A)$ (recall that the range of A and the kernel of A^T are orthogonal subspaces). The algorithm is summarized below.

Algorithm CGD: Conjugate gradient descent to compute $A^{\dagger}\varphi$

- Initialize $k \leftarrow 0$, $\psi_0 \leftarrow 0$, $r_0 \leftarrow A^T\varphi - A^TA\psi_0$, $d_0 \leftarrow r_0$.
- While $\|r_k\| > \varepsilon$, do
  - $\alpha_k = \frac{\|r_k\|^2}{d_k^TA^TAd_k}$
  - $\psi_{k+1} \leftarrow \psi_k + \alpha_k d_k$
  - $r_{k+1} \leftarrow r_k - \alpha_k A^TAd_k$
  - $d_{k+1} \leftarrow r_{k+1} + \frac{\|r_{k+1}\|^2}{\|r_k\|^2}d_k$
  - $k \leftarrow k+1$
- Return ψ_k

Notice that in our case where A is symmetric, this Algorithm CGD is nothing but the classical algorithm for solving $A^2\psi = A\varphi$. In this case, the range and kernel of A are orthogonal subspaces so that the convergence of the algorithm follows from the non-singular case (applied to the restriction of A^2 to the range of A).

Since the multiplication by A can be computed efficiently with the FFT, the complexity of Algorithm CGD with N iterations is $\mathcal{O}(N|\Omega|\log|\Omega|)$. The main benefit of using this algorithm is that it allows to consider very large conditioning sets $\mathcal{C}$.

Of course, increasing $\mathcal{C}$ may increase the number of required iterations to obtain the solution at a given precision ε. But if the condition number of the system is low, we will get a good approximation of the solution in a reasonable number of iterations. Let us mention that Algorithm CGD is theoretically expected to get the exact solution in a finite number of iterations, but this remark is not useful for our practical case because of the numerical errors caused by the FFT.

Stopping criterion. The stopping criterion that we use in Algorithm CGD is $\|r_k\| \leq \varepsilon$ where the residual at iteration k is given by

$$r_k = A^T\varphi - A^T A\psi_k, \tag{13}$$

and where $\|r_k\|$ is the unnormalized ℓ^2-norm of $r_k \in \mathbb{R}^{|\mathcal{C}|}$. In practice, to keep a simple choice, we take $\varepsilon := 10^{-3}$ and we also constrain the number of iterations to be less than $k_{\max} = 1000$. The numerical behavior of this CGD algorithm is studied in the IPOL companion paper.

3.4. Comments on the Kriging System.

The matrix A is not necessarily invertible. Indeed, let us consider the case of a color periodic ADSN model on Ω estimated by (9). Then the DFT of the covariance operator Γ is given by

$$\widehat{t}_v(\xi)\widehat{t}_v(\xi)^* = \begin{cases} \frac{1}{|\omega|}\widehat{v}(\xi)\widehat{v}(\xi)^* & \text{if } \xi \neq 0 \\ 0 & \text{if } \xi = 0 \end{cases}. \tag{14}$$

As noted in [86], this matrix has rank ≤ 1 which constrains the rank of the matrix Γ (of size $d|\Omega| \times d|\Omega|$) to be bounded by $|\Omega| - 1$. Since A is a submatrix of Ω, $\mathrm{Rank}(A) \leq |\Omega| - 1$. In particular, if the conditioning set is sufficiently big so that $d\,|\mathcal{C}| \geq |\Omega|$, then A cannot be invertible.

The vector $\varphi = u_{|\mathcal{C}} - \bar{u}$ may not be in the range of A. Indeed, if A is not invertible, the conditioning values could be out of the range of A. However this is not a problem to apply Algorithm CGD because taking $A\varphi$ implicitly cancels the component on the kernel of A.

Notice also that if the estimated ADSN model is well adapted to the masked texture, then it is likely that φ is close to the range of A. In practice, the distance of φ to the range of A is bounded by the norm of the residual obtained with the direct conjugate gradient method $\|\varphi - A\psi_k\| \geq \mathrm{dist}\big(\varphi, \mathrm{Range}(A)\big)$.

3.5. Complete Algorithm.

To end this section, we summarize our microtexture inpainting algorithm. In Algorithm CGD the matrix $A = \Gamma_{|\mathcal{C}\times\mathcal{C}}$ is not formed explicitly, and we only need to apply it efficiently with the FFT-based algorithm. Also, if one is not interested in the kriging and innovation components but only in the inpainting result, then only one instance of gradient descent is needed since the output only depends on $(u - \bar{v} - F)^* = \Gamma^\dagger_{|\mathcal{C}\times\mathcal{C}}(u_{|\mathcal{C}} - \bar{v} - F_{|\mathcal{C}})$.

The overall complexity of this algorithm is $\mathcal{O}(k_{\max}|\Omega|\log|\Omega|)$ where $k_{\max}$ is the number of iterations used in the gradient descent algorithm. The overall number of FFTs required by the whole inpainting process (whose detailed computation can be found in the IPOL companion paper) is $(4k_{\max} + 6)d$ FFTs. Using our C implementation (involving parallel computing, in particular for the FFT) run with a modern computer (Intel i7 processor @2.60GHz with 4 cores), the whole inpainting process takes about 20 seconds for a 256×256 and 1000 iterations of CGD.

Algorithm: Microtexture inpainting

Input: Mask $M \subset \Omega$, texture u on $\Omega \setminus M$, conditioning points $\mathcal{C} = \partial_3 M$.

- Choose a subdomain $\omega \subset \Omega \setminus M$ for the estimation (by default, $\omega = \Omega \setminus M$)
- From the restriction v of u to ω, compute

$$\bar{v} = \frac{1}{|\omega|} \sum_{x \in \omega} v(x), \qquad t_v = \frac{1}{\sqrt{|\omega|}} (v - \bar{v}) \mathbf{1}_\omega$$

- Draw a Gaussian sample $F = t_v * W$
- Compute $\psi_1 = \Gamma^\dagger_{|\mathcal{C}\times\mathcal{C}}(u_{|\mathcal{C}} - \bar{v})$, $\psi_2 = \Gamma^\dagger_{|\mathcal{C}\times\mathcal{C}} F_{|\mathcal{C}}$

(Algorithm CGD with $A = \Gamma_{|\mathcal{C}\times\mathcal{C}}$, $\varepsilon = 10^{-3}$ and $k_{\max} = 1000$ iterations)

- Extend ψ_1 and ψ_2 by zero-padding to get Ψ_1 and Ψ_2
- Compute

$$(u - \bar{v})^* = t_v * \tilde{t}_v^T * \Psi_1 \quad \text{(kriging component)}$$
$$F^* = t_v * \tilde{t}_v^T * \Psi_2 \quad \text{(innovation component)}$$

Output: Fill M with the values of $\bar{v} + (u - \bar{v})^* + F - F^*$

4. Results and Discussion.

4.1. Inpainting with an Oracle Model. First, we propose a validation experiment to confirm that Gaussian conditional simulation can be applied to constrained microtexture synthesis. For that, we consider a non-masked texture image u on which we estimate an oracle ADSN model as explained in Section 3.2. We compute one realization of this oracle ADSN model (with a random seed s_1), on which we put a mask M. Then we perform conditional sampling of the values in the masked region (with a random seed $s_2 \neq s_1$), based on a set of conditioning points $\mathcal{C}$, which is taken to be either $\mathcal{C} = \Omega \setminus M$ or $\mathcal{C} = \partial_3 M$. This amounts to applying our inpainting algorithm, except that we use an oracle model.

The results are reported in Fig. 4 for a square mask and in Fig. 5 for more irregular masks (obtained as level sets of white or correlated noise). Notice that in all these experiments, the result is visually perfect, in the sense that the inpainted texture is visually similar to a realization of the global ADSN model. Therefore, with our conjugate gradient descent scheme, the error made in the resolution of the linear system has only a negligible visual impact. Another important point raised by the results of Fig. 4 is that conditioning on the two different sets $\mathcal{C} = \Omega \setminus M$ and $\mathcal{C} = \partial_3 \Omega$ give very similar results. This illustrates that this inpainting scheme truly respects the covariance structure (and in particular the long-range correlations) even if the conditioning border is thin. Increasing further the conditioning border only adds some redundancy in the conditional model (and worsens the kriging system condition number). See Section 4.4 for a more detailed analysis of this parameter.

Let us remark that the results obtained in Fig. 5 with irregular masks look impressive at first sight since a wide majority of pixels are masked; but one should recall that in this experiment the oracle ADSN model is estimated on the unmasked exemplar, which makes the inpainting problem much simpler (compare with the results of Section 4.2).

In the experiment of Fig. 6, we show that Gaussian conditional simulation with an

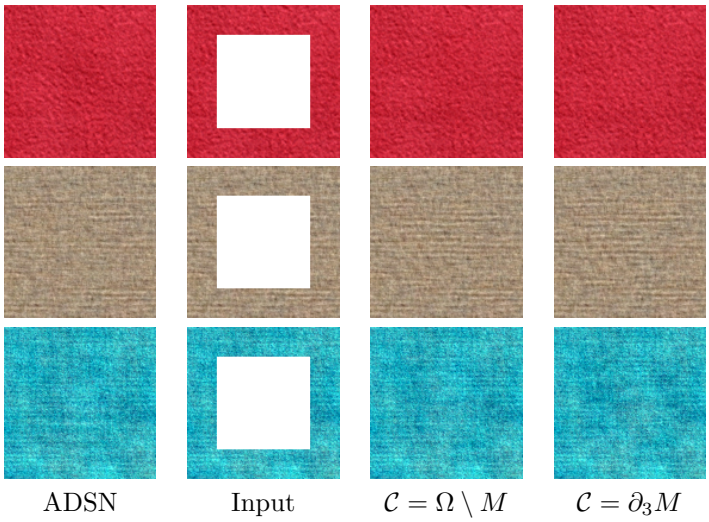

FIG. 4. ***Inpainting Gaussian textures with the oracle Gaussian model - regular masks.*** *The masked input has been inpainted with Gaussian conditional simulation using an oracle Gaussian model (estimated from the unmasked exemplar texture) based on conditioning values on $\mathcal{C} \subset \Omega$. From left to right, we show a sample of the oracle model, the masked input, and the inpainted results obtained for $\mathcal{C} = \Omega \setminus M$ or $\mathcal{C} = \partial_3 M$. The inpainted results are visually perfect in the sense that they cannot be distinguished from a sample of the oracle model. This is true both for $\mathcal{C} = \Omega \setminus M$ and $\mathcal{C} = \partial_3 M$ which shows that conditioning on $\mathcal{C} = \partial_3 M$ is practically sufficient.*

oracle model can be used to extrapolate textural content defined on a thin domain. In this case, the simulated conditional Gaussian vector is very high-dimensional, which illustrates the benefit of having a scalable algorithm based on gradient descent (and not on explicit computation of the covariance operators).

4.2. Inpainting with an Estimated Gaussian Model. In this section, we provide experimental results which show that our algorithm is able to inpaint holes in microtextures, whatever the size of the hole, and with only minimal requirements on the hole regularity. In contrast with the last section, the Gaussian model is now estimated from the masked exemplar. We will show that the naive estimation technique explained in Section 3.2 and illustrated in Fig. 3 leads to satisfying inpainting results except in the case where the mask is made of randomly scattered pixels. In the experiments shown in this section, we took $\mathcal{C} = \partial_3 M$.

In Fig. 7, we show some results of our algorithm for several microtextures and macrotextures, with various types of masks. As one can observe, the results with microtextures are globally very satisfying; the most difficult case being the irregular mask of the third column, for which the Gaussian model cannot be properly estimated, in accordance with one of the conclusions drawn in [66]. Surprisingly, we also obtained quite convincing results on more structured textures.

To end this section, we show that our algorithm can be used to inpaint textural parts of more general images. For example, on Fig. 8, we used it to remove some undesirable details located in a region composed of one homogeneous microtexture. In such a case, one must manually specify the subdomain ω on which the Gaussian model is estimated in order to take only values in the desired texture region.

4.3. Computing and Visualizing the Kriging Coefficients. In order to better understand the conditional simulation, it is interesting to visualize the kriging

FIG. 5. ***Inpainting Gaussian textures with the oracle Gaussian model - irregular masks.*** *The masked input has been inpainted with Gaussian conditional simulation using an oracle Gaussian model (estimated from the unmasked exemplar texture) based on conditioning values on* $\mathcal{C} \subset \Omega$*. From left to right, we display a sample of the oracle model, a first masked input (the mask is obtained as an excursion set of a Gaussian process) and the corresponding inpainting result, and a second masked input (the pixels are masked independently with probability* 0.8*). Again, these inpainted results are visually perfect since they look exactly like a realization of the global ADSN model.*

FIG. 6. ***Gaussian texture extrapolation with an oracle Gaussian model.*** *From left to right: input images, extrapolated texture ($\mathcal{C} = \partial_3 M$), baseline result (obtained with an independent ADSN realization on the mask). The images are of size 621×427. The extrapolation by Gaussian conditional simulation has succeeded since the letters cannot be retrieved in the resulting image. In contrast, with the baseline method, the border of the extrapolated region is still visible (essentially because of the low frequency component).*

coefficients. Heuristically speaking, every non-zero coefficient $\lambda_c(x)$ corresponds to a position x whose value $F(x)$ depends on $F(c)$ in the conditional simulation. We can thus expect the correlations of the adopted Gaussian model to be reflected in the kriging coefficients.

First, let us explain how to visualize $(\lambda_c(x))_{x\in\Omega}$ for a fixed $c \in \mathcal{C}$. We have

$$(\lambda_c(x))_{x\in\Omega} = \Lambda\delta_c = \Gamma_{|\Omega\times\mathcal{C}}\Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}\delta_c, \tag{15}$$

where we used the notation $\delta_c = (\mathbf{1}_{c=d})_{d\in\mathcal{C}}$. Thus, to compute $(\lambda_c(x))_{c\in\mathcal{C}}$, we just use our algorithm on a Dirac input.

In a dual manner, one can also visualize $(\lambda_c(x))_{c\in\mathcal{C}}$ for each $x \in \Omega$. For that, we simply notice that

$$(\lambda_c(x))_{c\in\mathcal{C}} = \Lambda^T\delta_x = \Gamma^{\dagger}_{|\mathcal{C}\times\mathcal{C}}\Gamma_{|\mathcal{C}\times\Omega}\delta_x, \tag{16}$$

where $\delta_x = (\mathbf{1}_{x=y})_{y\in\Omega}$. So the computation of these coefficients can be done in a similar fashion, except that the covariance convolution $\Gamma_{|\mathcal{C}\times\Omega}$ is performed before pseudo-inverse computation (with Algorithm CGD).

In the case of the inpainting application, we get the coefficients shown in Fig. 9. These results clearly indicate that the correlations captured in the Gaussian model are reflected by the large kriging coefficients. We can also observe on this figure that the kriging coefficients are not positive in general.

4.4. Impact of the Size of the Conditioning Border. In this section, we investigate the impact of changing the size of the conditioning border. Again, an ideal setting would be to choose $\mathcal{C} = \Omega \setminus M$, but then the kriging system is very large. Here we will confirm that taking $\mathcal{C} = \partial_w M$ is sufficient, and we will precisely examine the variation of the conditional model when increasing the width w of the border.

In order to give a quantitative comparison, we suggest to compute distances between the conditional models, which are basically Gaussian random vectors on M.

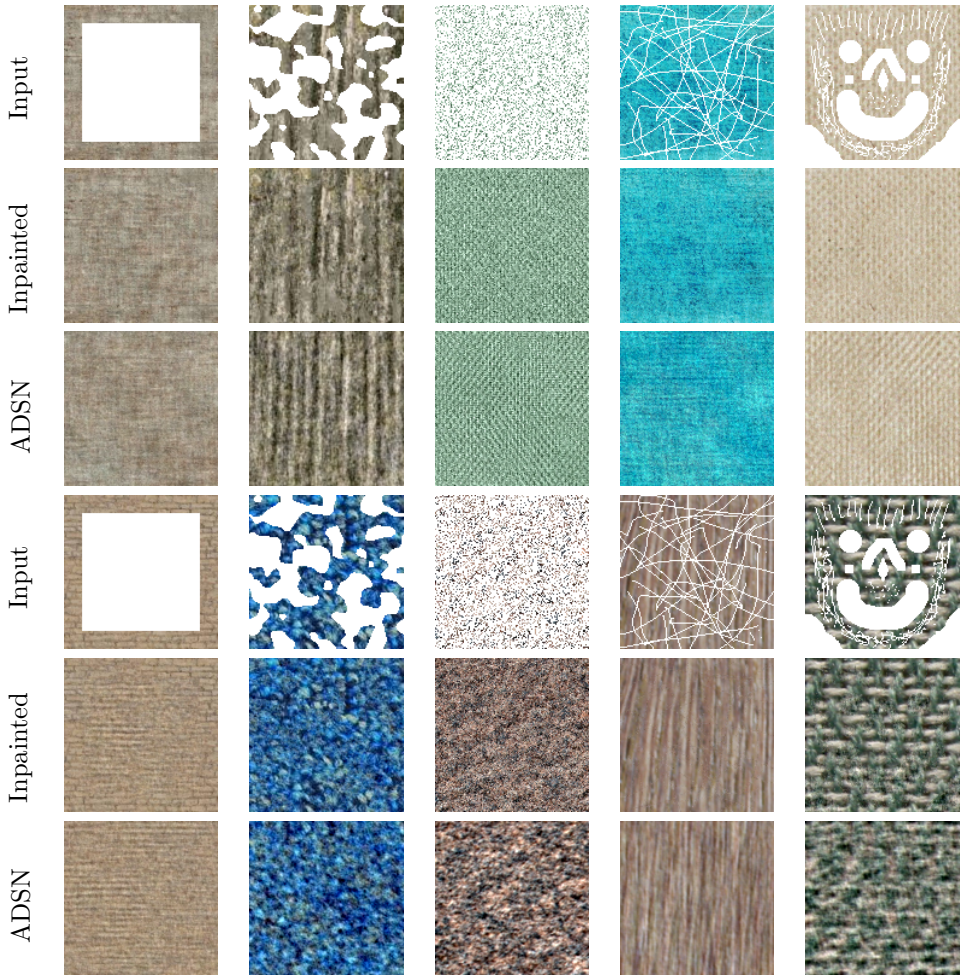

FIG. 7. ***Examples of textural inpainting.*** *We present results of our inpainting method for several textures and masks. From top to bottom (rows 1-3 and rows 4-6), we display a masked input, the inpainted result, and a sample of the estimated ADSN model (which is useful to exhibit the limit of the Gaussian model). On rows 1-3, we display results on microtextures, while on rows 4-6 we display results on more structured textures. The results on microtextures are visually pleasing, except for the irregular mask of the third column. The results on macrotextures are of course not as perfect (in particular, for the wood example of the bottom of fourth column, the mask is still visible on close examination). Nevertheless, it is surprising that our method (based on Gaussian synthesis) still gives convincing results on some macrotextures.*

A possible way to perform this comparison is to rely on the L^2-optimal transport distance, which has already been used in several works about texture synthesis [86, 38]. Let us recall [29] that the L^2-optimal transport distance between two Gaussian models $\mu_X = \mathcal{N}(m_X, \Sigma_X), \mu_Y = \mathcal{N}(m_Y, \Sigma_Y)$ is given by

$$d_{\mathrm{OT}}(\mu_X, \mu_Y)^2 = \|m_X - m_Y\|^2 + \mathrm{Tr}(\Sigma_X) + \mathrm{Tr}(\Sigma_Y) - 2\mathrm{Tr}\Big((\Sigma_X \Sigma_Y)^{1/2}\Big). \tag{17}$$

We consider a gray-level exemplar texture $u : \Omega \to \mathbb{R}$ on which we estimate an oracle model $\mathcal{N}(\bar{u}, \Gamma)$ and on which we put a mask $M \subset \Omega$. Then, we consider the reference conditional model $\mu_\infty = \mathcal{N}(m_\infty, \Sigma_\infty)$ obtained with $\mathcal{C}_\infty = \Omega \setminus M$, and the

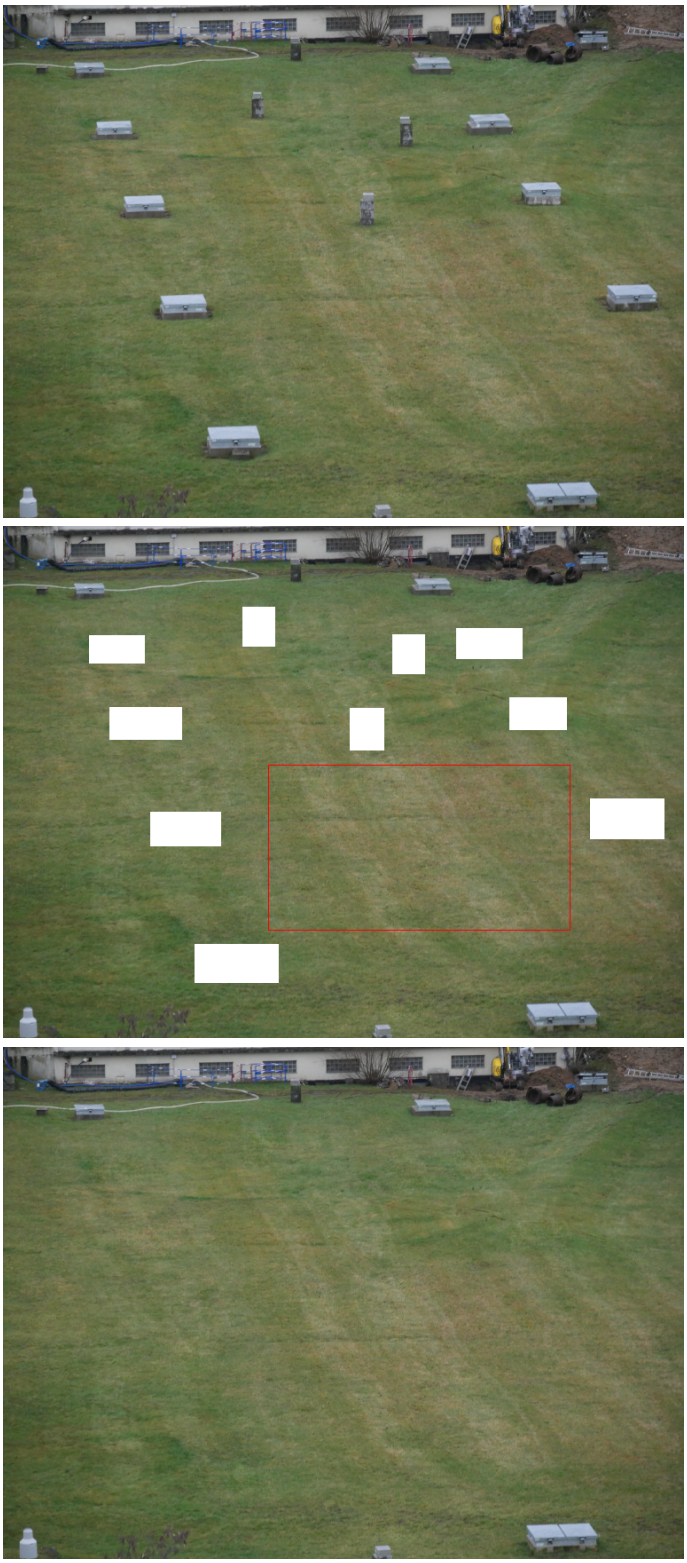

FIG. 8. ***Inpainting textural parts of an image.*** *From top to bottom, we display the original image (of size* 768×577*), the masked input (the Gaussian model has been estimated in the sub-domain* ω *delimited by the red box), and the inpainted result. Our algorithm is able to synthesize microtexture content which naturally blends with the surrounding context.*

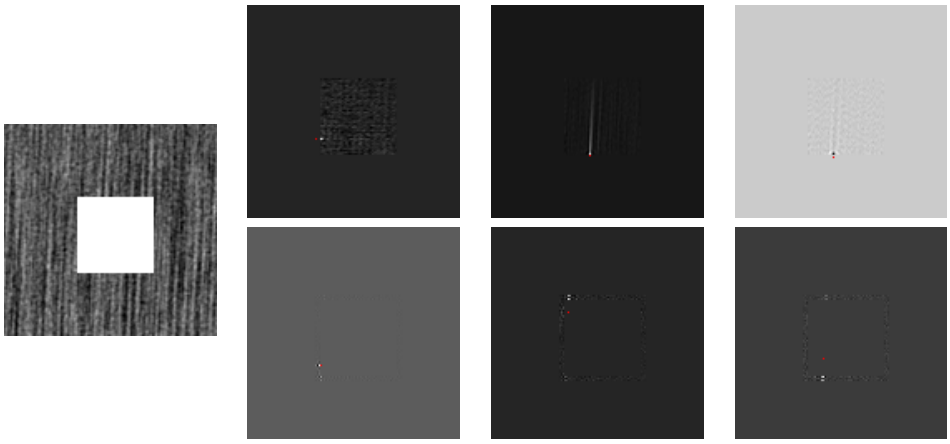

FIG. 9. ***Visualizing Kriging coefficients.*** *In the first column, we display the masked input. For the three other columns: in the first row, we display the kriging coefficients* $(\lambda_c(x))_{x\in M}$ *for different positions of the conditioning pixel* $c \in \mathcal{C}$ *(drawn in red); in the second row, we display the kriging coefficients* $(\lambda_c(x))_{c\in\mathcal{C}}$ *for different positions of the pixel* $x \in M$ *(drawn in red). So in the first row, we can observe the values that will be more impacted by a given conditioning point* c*, and in the second row, we can observe the conditioning values which contribute most in conditional sampling at a given position* x*. The kriging coefficients are obtained from an oracle model estimated on the unmasked exemplar and we took* $\mathcal{C} = \partial_3 M$*. The color map is renormalized in each case. It is interesting to remark that the vertical correlations captured by this texture model are reflected by larger kriging coefficients.*

conditional models $\mu_w = \mathcal{N}(m_w, \Sigma_w)$ obtained with $\mathcal{C}_w = \partial_w M$ (border of M with width w pixels). Using the expressions found in Section 2.3 and Section 2.4, we recall

$$m_w = \Gamma_{|M\times\mathcal{C}_w}\Gamma^{\dagger}_{|\mathcal{C}_w\times\mathcal{C}_w}(u-\bar{u})_{|\mathcal{C}_w}, \quad \Sigma_w = \Gamma_{|M\times M} - \Gamma_{|M\times\mathcal{C}_w}\Gamma^{\dagger}_{|\mathcal{C}_w\times\mathcal{C}_w}\Gamma_{|\mathcal{C}_w\times M}.$$

For our experiment, we choose a reasonably small texture so that all these covariance matrices can be explicitly built and stored (relying on standard numerical routines for pseudo-inverse and square roots computation[2]). We then plot the function

$$w \in \{1,\dots,20\} \longmapsto \frac{d_{\mathrm{OT}}(\mu_w,\mu_\infty)}{\sigma_u\sqrt{|M|}}\ , \tag{18}$$

where σ_u is the marginal standard deviation of the oracle model. We also report separately the distances between the mean values and the covariance matrices, i.e.

$$d(m_w,m_\infty) = \|m_w - m_\infty\|, \quad d(\Sigma_w,\Sigma_\infty)^2 = \mathrm{Tr}(\Sigma_w) + \mathrm{Tr}(\Sigma_\infty) - 2\mathrm{Tr}\Big((\Sigma_w\Sigma_\infty)^{1/2}\Big).$$

The results can be observed in Fig 10. One can observe a global tendency of these distances to decrease when the conditioning border gets larger. But we do not observe a sudden plunge of the value (even if the covariance distance decreases a bit quicker for $w < 5$). Also, an interesting fact raised by these graphs is that the marginal error made when replacing $\mathcal{C}_\infty$ by $\mathcal{C}_w$ is in general less than one σ_u. Notice also that when w increases, the kriging system become more and more ill-conditioned.

[2]The pseudo-inverse is only computed up to a given precision. But, following the remark at the end of Section 3.4, we checked that after conditional simulation with the approximate kriging coefficients, the covariance matrix of the global Gaussian model is the desired one up to an error of ℓ^∞-norm less than 10^{-15}.

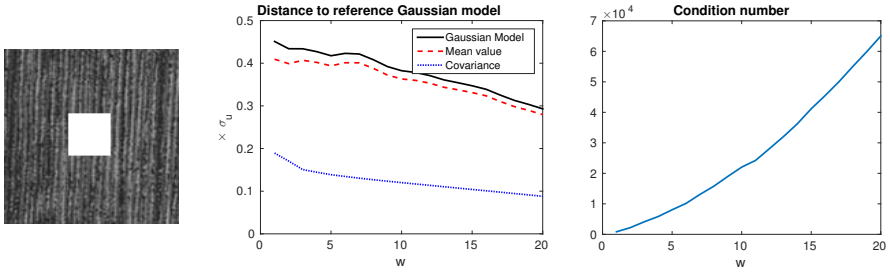

FIG. 10. ***Quantative study of the conditional models depending on the conditioning set $\mathcal{C}$.*** *We computed the distance between the reference conditional model (obtained for $\mathcal{C}_\infty = \Omega \setminus M$) and the conditional models (obtained for $\mathcal{C}_w = \partial_w M$), see (18). On the same diagram, we also show the distance between the mean and covariance components separately. On the right diagram, we display the conditioning number of the kriging system. When w increases, the conditional model slowly gets closer to the reference model, and the conditioning number increases.*

We also propose in Fig. 11 a more qualitative experiment. This qualitative study is important to examine the quality of the inpainting result around the mask border (which is not reflected through the marginal L^2 error between two conditional models). For several values of the border width $w = 1, 3, 5$, we inpaint a texture image (with the oracle Gaussian model), and we compare the results with the one obtained in the ideal case $\mathcal{C}_\infty = \Omega \setminus M$. In order to give per-pixel comparison, we used the same random seed for the conditional sampling. Apart from the visual results, we also report the distance between the mean values of the corresponding conditional models.

It is interesting to notice that the kriging components look very different with $w = 5$ and $w = \infty$. Indeed, when the conditioning set gets larger, the kriging component depends on a larger number of random variables, and thus has an increased stochastic nature. This explains why the distance between the Gaussian models (or their mean or covariance functions) does not quickly tend to zero when w increases. Still, as reflected by the example of Fig. 11 and as observed in all our experiments, the inpainting result is already good for $w = 3$ (in particular, for many textures, this value is sufficient to naturally blend the inpainted domain in the context).

To conclude this section, we confirm that taking $\mathcal{C} = \partial_3 M$ is in general sufficient for our inpainting purpose. Besides, growing $\mathcal{C}$ adds redundancy in the kriging system, and also increases the stochastic nature of the kriging component.

4.5. Comparisons. In this section, we compare our microtexture inpainting algorithm with several recent inpainting techniques.

First, in Fig. 12, we compare our method with two very famous methods, namely, total variation (TV) based inpainting [18], and the patch-based method of Criminisi et al [23]. As could be expected, the TV inpainting method is not appropriate for this example, because the water texture in this image is not of bounded variation. In contrast, much better results are obtained with our method or the one of Criminisi et al. Compared to [23], our result seems a bit more stochastic, maybe even too stochastic in the upper part of the inpainted domain. This clearly reflects one limitation of our model, which is stationarity.

On Fig. 13 and Fig. 14, we compare our Gaussian inpainting algorithm with several patch-based methods. On the first rows of Fig. 13, one can observe that Gaussian inpainting gives nearly perfect results on microtextures (which was expected). Also, the last rows of Fig. 13 show that the results obtained on macrotextures, although

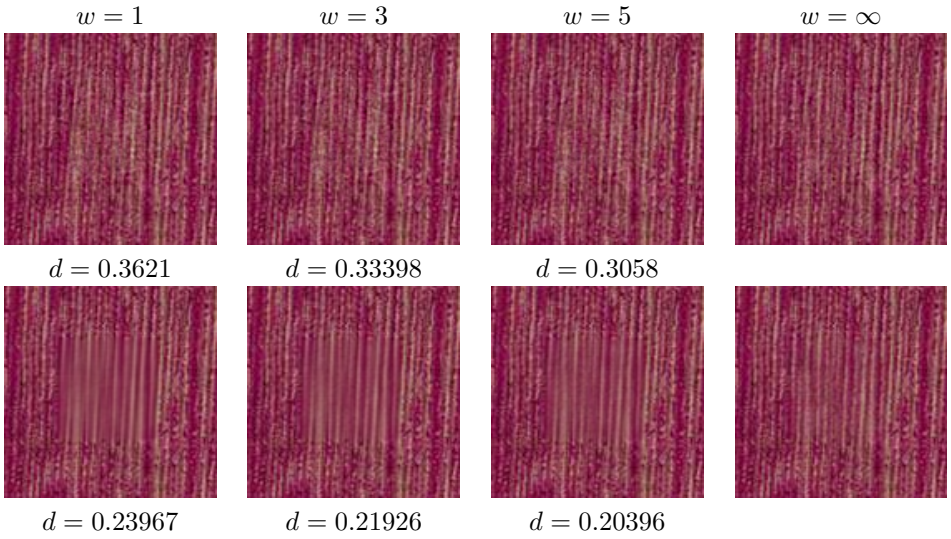

FIG. 11. ***Qualitative study of the conditional models depending on the conditioning set $\mathcal{C}$.*** *From left to right, we display the inpainting results obtained for $\mathcal{C}$ being a border of M of width $w = 1, 3, 5$ pixels, and also the limit solution $\mathcal{C} = \Omega \setminus M$. In the first row, we display the sample of the conditional model, and on the second row the mean value of the conditional model (kriging component). In both rows, we compute the standard ℓ^2-distance to the image shown on the right (normalized by $\sigma_u \sqrt{|M|d}$). See the text for additional comments.*

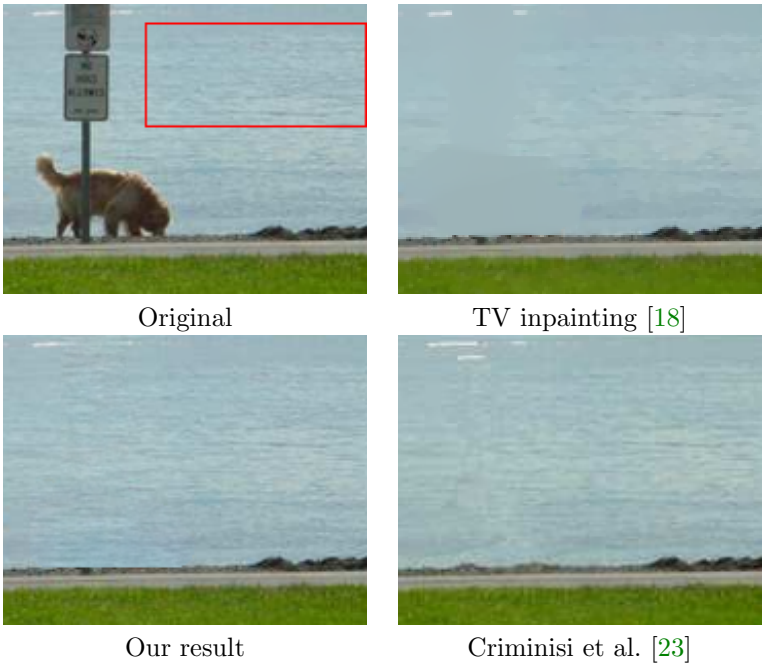

FIG. 12. ***Comparison with [18, 23].*** *In the first row, we display the original image (taken from [23]) on the left, and on the right the result of TV inpainting [18] (obtained with the implementation available at [41]). In the second row, on the left we show the result of Gaussian inpainting (with a model estimated in the red box), and on the right the result of the patch-based method of [23]. As one can see, the TV inpainting is not able to preserve texture. In contrast, the method of [23] is truly able to generate textural content, but may lead to repetition artifacts.*

not perfect, are still quite convincing in comparison to patch-based methods. Even if Gaussian inpainting is not able to preserve salient geometric features, it has two important benefits: the synthesized content is smoothly blended in the input data, and the synthesized content does not suffer from repetition artifacts. But of course, Gaussian inpainting will clearly fail if one tries to inpaint a very thick hole in a highly non Gaussian texture (because the human visual system is able to discriminate between a highly structured texture and its ADSN counterpart). Let us mention that some examples of Fig. 13 are difficult to handle with patch-based methods because the number of available patches in the unmasked area is quite small, which favors repetitions. This is a noticeable advantage of our method to be applicable even if the unmasked part does not contain many complete patches.

All these remarks are confirmed with the results of Fig. 14 which provides a comparison of these methods on a difficult textural inpainting problem. This striking example clearly exhibits the benefits and drawbacks of each method. With Gaussian inpainting, the color distribution and frequential content are precisely respected, and long-range correlations are preserved (as can be seen in the kriging component), but complex geometric structures are not properly synthesized as they would be with a patch-based method. In contrast, with patch-based methods, even if there is enough available patches here, we observe some repetition artifacts which can be explained in the same way as the *growing garbage* effect which was already brought up by the seminal paper [31]. There may also be other artifacts which are more specific: on the result of [3], the inpainted domain is a bit too blurry and the border of the inpainted domain is still clearly visible; and on the result of [69], after close examination of the inpainted domain, we can perceive small seams which are due to changes in the offsets used for region pasting.

5. Conclusion. In this paper, we proposed a stochastic inpainting method based on Gaussian conditional simulation. It is able to inpaint holes of any shape and size in microtexture images while precisely respecting a random texture model. Gaussian texture inpainting shares of course some limitations with Gaussian texture synthesis, but we have illustrated on many texture images that this simple approach competes with state-of-the-art inpainting algorithms in terms of visual results.

As discussed in the paper, we have proposed a very simple procedure for estimating a Gaussian texture model from a masked exemplar texture. Numerical experiments show that this naive technique gives good results provided that the mask complement contains a sufficiently plain piece of texture. Still, we believe that it would be interesting to dispose of a more robust estimation technique amenable to deal with very irregular masks. This may be rephrased as parameter estimation with hidden variables and might be addressed with an expectation-maximization technique, but keeping the computational cost of such a procedure seems very challenging. Notice that this problem has already been generally discussed in [79] and more particularly addressed in [56, 27, 73] in a Bayesian framework for parametrized covariances.

A promising (but equally challenging) direction for future work is to extend conditional simulation to non-stationary models in order to address inpainting of images of natural scenes. It is likely that for such images, one should use a deterministic method for extension of geometric structures, coupled with a (conditional) stochastic step to complete the textural content. Such a model would build another bridge between variational and stochastic inpainting, thus shedding another light on the question whether inpainting should be considered as minimizing a functional or sampling a large-scale distribution.

FIG. 13. ***Comparison with patch-based methods (I).*** *On each row, from left to right, we display a masked input, the result of our Gaussian inpainting algorithm, the result of [69], the result of variational non-local inpainting [3] (obtained with the online implementation of [35] using the NLmeans option), and the result of [15] (obtained with the publicly available* G'MIC *plugin for* GIMP *[82]). With the results of the fourth first rows, one clearly sees that Gaussian inpainting gives much better results on microtextures. The results of the last rows show that Gaussian inpainting also gives reasonable results on macrotextures, and in particular, it avoids the repetition artifacts that can sometimes be encountered with patch-based synthesis (first and fifth rows). In contrast patch-based inpainting better preserves geometric features (like the stitches of the sixth and seventh examples) which are completely lost with Gaussian synthesis.*

FIG. 14. ***Comparison with patch-based methods (II).*** *We compare several inpainting methods on a difficult textural inpainting problem. On the first row, from left or right, we display the masked input, the result of our method, together with the corresponding kriging component. On the second row, we display the results of variational non-local inpainting [3] (obtained with the online implementation of [35] using the NLmeans option), the result of [15] (obtained with the publicly available* G'MIC *plugin for* GIMP *[82]), and the result of [69]. Again, we observe on this example that Gaussian inpainting fills the hole with a truly stochastic content which respects the second-order statistic of the texture (in particular the color distribution and the power spectrum), but fails to reproduce the geometric features in contrast to patch-based methods. The second row precisely highlights typical artifacts associated with state-of-the-art patch-based methods: with [3] the inpainted content is too blurry; with [15] we get repetition artifacts; and with [69] we can perceive small seams between inpainted regions using different offsets.*

6. Acknowledgments. This work has been partially funded by the French Research Agency (ANR) under grant nro ANR-14-CE27-0019 (MIRIAM).

We thank Alasdair Newson for his comments and for providing us with the implementation of [69]. We thank Olivier le Meur, Lionel Moisan, and Frédéric Richard for several discussions about inpainting or kriging estimation. Finally, we thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] A. ALMANSA, F. CAO, Y. GOUSSEAU, AND B. ROUGÉ, *Interpolation of digital elevation models using AMLE and related methods*, IEEE transactions on geoscience and remote sensing, 40 (2002), pp. 314–325.

[2] P. ARIAS, V. CASELLES, AND G. FACCIOLO, *Analysis of a Variational Framework for Exemplar-Based Image Inpainting*, Multiscale Modeling & Simulation, 10 (2012), pp. 473–514, doi:10.1137/110848281.

[3] P. ARIAS, G. FACCIOLO, V. CASELLES, AND G. SAPIRO, *A variational framework for exemplar-based image inpainting*, International Journal of Computer Vision, 93 (2011), pp. 319–347.

[4] J. AUJOL, S. LADJAL, AND S. MASNOU, *Exemplar-based inpainting from a variational point of view*, SIAM Journal on Mathematical Analysis, 42 (2010), pp. 1246–1285.

[5] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO, AND J. VERDERA, *Filling-in by joint*

interpolation of vector fields and graylevels, IEEE Transactions on Image Processing, 10 (2001), pp. 1200–1211.

[6] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, *PatchMatch: a randomized correspondence algorithm for structural image editing*, ACM Transactions on Graphics, 28 (2009).

[7] M. Bertalmio, A. Bertozzi, and G. Sapiro, *Navier-stokes, fluid dynamics, and image and video inpainting*, in Proceedings of CVPR, vol. 1, IEEE, 2001.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image Inpainting*, in Proc. of SIGGRAPH, 2000, pp. 417–424, doi:10.1145/344779.344972.

[9] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, *Simultaneous structure and texture image inpainting*, IEEE Transactions on Image Processing, 12 (2003), pp. 882–889.

[10] A. Bertozzi, S. Esedoglu, and A. Gillette, *Inpainting of binary images using the Cahn-Hilliard equation*, IEEE Transactions on image processing, 16 (2007), pp. 285–291.

[11] R. Bornard, E. Lecan, L. Laborelli, and J. Chenot, *Missing data correction in still images and image sequences*, in Proceedings of the tenth ACM international conference on Multimedia, 2002, pp. 355–361.

[12] F. Bornemann and T. März, *Fast image inpainting based on coherence transport*, Journal of Mathematical Imaging and Vision, 28 (2007), pp. 259–278.

[13] A. Bugeau, M. Bertalmo, V. Caselles, and G. Sapiro, *A comprehensive framework for image inpainting*, IEEE Transactions on Image Processing, 19 (2010), pp. 2634–2645.

[14] M. Burger, L. He, and C. Schönlieb, *Cahn-Hilliard inpainting and a generalization for grayvalue images*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 1129–1167.

[15] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray, *Exemplar-based Inpainting: Technical Review and new Heuristics for better Geometric Reconstructions*, IEEE Transactions on Image Processing, 24 (2015), pp. 1809–1824.

[16] J. Cai, R. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Applied and Computational Harmonic Analysis, 24 (2008), pp. 131–149.

[17] F. Cao, Y. Gousseau, S. Masnou, and P. Pérez, *Geometrically guided exemplar-based inpainting*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 1143–1179.

[18] T. Chan and J. Shen, *Mathematical models for local nontexture inpaintings*, SIAM Journal on Applied Mathematics, 62 (2002), pp. 1019–1043.

[19] S. Chandra, M. Petrou, and R. Piroddi, *Texture Interpolation Using Ordinary Kriging*, Pattern Recognition and Image Analysis, (2005), pp. 183–190.

[20] J. Chilès, *Quelques méthodes de simulation de fonctions aléatoires intrinsèques*, Cahiers de Géostatistique, 5 (1995), pp. 97–112.

[21] J.-P. Chilès and P. Delfiner, *Geostatistics: modeling spatial uncertainty*, John Wiley & Sons, 2009.

[22] N. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, 1993.

[23] A. Criminisi, P. Pérez, and K. Toyama, *Region filling and object removal by exemplar-based image inpainting*, IEEE Transactions on Image Processing, 13 (2004), pp. 1200–1212.

[24] L. Demanet, B. Song, and T. Chan, *Image inpainting by correspondence maps: a deterministic approach*, Applied and Computational Mathematics, 1100 (2003), p. 99.

[25] A. Desolneux, L. Moisan, and S. Ronsin, *A compact representation of random phase and Gaussian textures*, in Proceedings of ICASSP, 2012, pp. 1381–1384.

[26] C. Deutsch and A. Journel, *Geostatistical software library and user's guide*, vol. 119, Oxford University Press, New York, 1992.

[27] P. Diggle, J. Tawn, and R. Moyeed, *Model-based geostatistics*, Journal of the Royal Statistical Society: Series C (Applied Statistics), 47 (1998), pp. 299–325.

[28] J. Doob, *Stochastic processes*, Wiley, 1990.

[29] D. C. Dowson and B. V. Landau, *The Fréchet distance between multivariate normal distributions*, Journal of Multivariate Analysis, 12 (1982), pp. 450–455, doi:10.1016/0047-259X(82)90077-X.

[30] I. Drori, D. Cohen-Or, and H. Yeshurun, *Fragment-based image completion*, in ACM Transactions on Graphics, vol. 22, 2003, pp. 303–312.

[31] A. A. Efros and T. K. Leung, *Texture synthesis by non-parametric sampling*, in Proceedings of ICCV, vol. 2, 1999, pp. 1033–1038.

[32] M. Elad, J. L. Starck, P. Querre, and D. L. Donoho, *Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)*, Applied and Computational Harmonic Analysis, 19 (2005), pp. 340–358, doi:10.1016/j.acha.2005.03.005.

[33] X. Emery, *Conditioning Simulations of Gaussian Random Fields by Ordinary Kriging*, Mathematical Geology, 39 (2007), pp. 607–623, doi:10.1007/s11004-007-9112-x.

[34] S. Esedoglu and J. Shen, *Digital inpainting based on the Mumford-Shah-Euler image model*,

731 European Journal of Applied Mathematics, 13 (2002), pp. 353–370.
732 [35] V. Fedorov, G. Facciolo, and P. Arias, *Variational Framework for Non-Local Inpainting*,
733 Image Processing On Line, 5 (2015), pp. 362–386, doi:10.5201/ipol.2015.136.
734 [36] B. Galerne, Y. Gousseau, and J.-M. Morel, *Random Phase Textures: Theory and Syn-*
735 *thesis*, IEEE Transactions on Image Processing, 20 (2011), pp. 257–267, doi:10.1109/TIP.
736 2010.2052822.
737 [37] B. Galerne and A. Leclaire, *An Algorithm for Gaussian Texture Inpainting*, submitted to
738 Image Processing Online, (2017).
739 [38] B. Galerne, A. Leclaire, and L. Moisan, *A Texton for Fast and Flexible Gaussian Texture*
740 *Synthesis*, in Proceedings of EUSIPCO, 2014, pp. 1686–1690.
741 [39] B. Galerne, A. Leclaire, and L. Moisan, *Microtexture Inpainting Through Gaussian Con-*
742 *ditional Simulation*, in Proceedings of the International Conference on Acoustics, Speech
743 and Signal Processing (ICASSP), IEEE, 2016.
744 [40] B. Galerne, A. Leclaire, and L. Moisan, *Texton Noise*, Computer Graphics Forum, (2017),
745 doi:10.1111/cgf.13073.
746 [41] P. Getreuer, *Total Variation Inpainting using Split Bregman*, Image Processing On Line, 2
747 (2012), pp. 147–157, doi:10.5201/ipol.2012.g-tvi.
748 [42] J. Gómez-Hernàndez and E. Cassiraga, *Theory and Practice of Sequential Simulation*, in
749 Geostatistical Simulations, no. 7 in Quantitative Geology and Geostatistics, Springer, 1994,
750 pp. 111–124, doi:10.1007/978-94-015-8267-4_10.
751 [43] P. Goovaerts, *Geostatistics for Natural Resources Evaluation*, Oxford University Press, New
752 York, first edition ed., 1997.
753 [44] C. Guillemot and O. Le Meur, *Image inpainting: Overview and recent advances*, IEEE
754 Signal Processing Magazine, 31 (2014), pp. 127–144.
755 [45] A. Gutjahr, *Fast Fourier Transforms for Random Field Generation*, New Mexico Institute of
756 Mining and Technology, 1989.
757 [46] K. He and J. Sun, *Image completion approaches using the statistics of similar patches*, IEEE
758 Transactions on Pattern Analysis and Machine Intelligence, 36 (2014), pp. 2423–2435.
759 [47] D. Heeger and J. Bergen, *Pyramid-based texture analysis/synthesis*, in Proceedings of the
760 22nd annual conference on Computer graphics and interactive techniques, 1995, pp. 229–
761 238.
762 [48] Y. Hoffman and E. Ribak, *Constrained realizations of Gaussian fields: a simple algorithm*,
763 The Astrophysical Journal, 380 (1991), pp. L5–L8.
764 [49] H. Igehy and L. Pereira, *Image replacement through texture synthesis*, in Proceedings of
765 ICIP, vol. 3, 1997, pp. 186–189.
766 [50] J. Jia and C. Tang, *Inference of segmented color and texture description by tensor voting*,
767 IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 771–786,
768 doi:10.1109/TPAMI.2004.10.
769 [51] A. Journel, *Geostatistics for conditional simulation of ore bodies*, Economic Geology, 69
770 (1974), pp. 673–687.
771 [52] W. Kammerer and M. Nashed, *On the Convergence of the Conjugate Gradient Method*
772 *for Singular Linear Operator Equations*, SIAM Journal on Numerical Analysis, 9 (1972),
773 pp. 165–181, doi:10.1137/0709016.
774 [53] N. Komodakis and G. Tziritas, *Image completion using efficient belief propagation via prior-*
775 *ity scheduling and dynamic pruning*, IEEE Transactions on Image Processing, 16 (2007),
776 pp. 2649–2661.
777 [54] J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong, *Solid*
778 *texture synthesis from 2d exemplars*, ACM Transactions on Graphics, 26 (2007).
779 [55] C. Lantuéjoul, *Geostatistical Simulation: Models and Algorithms*, Springer, 2002.
780 [56] N. Le and J. Zidek, *Interpolation with uncertain spatial covariances: A Bayesian alterna-*
781 *tive to Kriging*, Journal of Multivariate Analysis, 43 (1992), pp. 351–374, doi:10.1016/
782 0047-259X(92)90040-M.
783 [57] O. Le Meur and C. Guillemot, *Super-resolution-based inpainting*, in ECCV 2012, Springer,
784 2012, pp. 554–567.
785 [58] M. Le Ravalec-Dupin, B. Noetinger, and L. Hu, *The FFT moving average (FFT-MA)*
786 *generator: An efficient numerical method for generating and conditioning Gaussian sim-*
787 *ulations*, Mathematical Geology, 32 (2000), pp. 701–723.
788 [59] A. Leclaire, *Random Phase Fields and Gaussian Fields for Image Sharpness Assessment and*
789 *Fast Texture Synthesis*, PhD thesis, Université Paris Descartes, 2015.
790 [60] J. Lewis, *Texture Synthesis for Digital Painting*, in Proceedings of SIGGRAPH, 1984, pp. 245–
791 252, doi:10.1145/800031.808605.
792 [61] J. Lewis, *Methods for Stochastic Spectral Synthesis*, in Proceedings on Graphics Interface,

1986, pp. 173–179.
[62] L. Li, T. Romary, and J. Caers, *Universal kriging with training images*, Spatial Statistics, (2015).
[63] Y. Liu and V. Caselles, *Exemplar-based image inpainting using multiscale graph cuts*, IEEE Transactions on Image Processing, 22 (2013), pp. 1699–1711.
[64] J. Mairal, M. Elad, and G. Sapiro, *Sparse Representation for Color Image Restoration*, IEEE Transactions on Image Processing, 17 (2008), pp. 53–69, doi:10.1109/TIP.2007.911828.
[65] G. Marietho z and S. Lefebvre, *Bridges between multiple-point geostatistics and texture synthesis: Review and guidelines for future research*, Computers & Geosciences, 66 (2014), pp. 66–80.
[66] G. Marietho z and P. Renard, *Reconstruction of incomplete data sets or images using direct sampling*, Mathematical Geosciences, 42 (2010), pp. 245–268.
[67] S. Masnou, *Disocclusion: a variational approach using level lines*, IEEE Transactions on Image Processing, 11 (2002), pp. 68–76.
[68] S. Masnou and J.-M. Morel, *Level lines based disocclusion*, in Proceedings of ICIP, vol. 3, 1998, pp. 259–263, doi:10.1109/ICIP.1998.999016.
[69] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, *Video Inpainting of Complex Scenes*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1993–2019, doi:10.1137/140954933.
[70] D. Oliver, *Moving averages for Gaussian simulation in two and three dimensions*, Mathematical Geology, 27 (1995), pp. 939–960, doi:10.1007/BF02091660.
[71] P. Pérez, M. Gangnet, and A. Blake, *Patchworks: Example-based region tiling for image editing*, Microsoft Research, MSR-TR-2004-04, Tech. Rep, (2004).
[72] G. Peyré, *Texture Synthesis with Grouplets*, IEEE Transactions on PAMI, 32 (2010), pp. 733–746, doi:10.1109/TPAMI.2009.54.
[73] J. Pilz and G. Spöck, *Why do we need and how should we implement Bayesian kriging methods*, Stochastic Environmental Research and Risk Assessment, 22 (2008), pp. 621–632.
[74] L. Raad, A. Desolneux, and J. Morel, *A Conditional Multiscale Locally Gaussian Texture Synthesis Algorithm*, Journal of Mathematical Imaging and Vision, (2016), pp. 1–20.
[75] L. Raad, A. Desolneux, and J.-M. Morel, *Conditional Gaussian Models for Texture Synthesis*, in Proceedings of Scale Space and Variational Methods in Computer Vision, 2015.
[76] H. Rue, *Fast sampling of Gaussian Markov random fields*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63 (2001), pp. 325–338.
[77] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, CRC Press, 2005.
[78] C. Schönlieb, *Partial Differential Equation Methods for Image Inpainting*, Cambridge University Press, 2015.
[79] M. Stein, *Interpolation of spatial data: some theory for kriging*, Springer, 1999.
[80] J. Sun, L. Yuan, J. Jia, and H. Shum, *Image completion with structure propagation*, in ACM Transactions on Graphics, vol. 24, ACM, 2005, pp. 861–868.
[81] A. Telea, *An image inpainting technique based on the fast marching method*, Journal of Graphics Tools, 9 (2004), pp. 23–34.
[82] D. Tschumperlé, *GREYC's Magic for Image Computing (GIMP plugin)*, http://gmic.eu/.
[83] J. J. Van Wijk, *Spot noise texture synthesis for data visualization*, in Proc. of SIGGRAPH, vol. 25, 1991, pp. 309–318.
[84] L. Wei and M. Levoy, *Fast texture synthesis using tree-structured vector quantization*, in Proceedings of SIGGRAPH, 2000, pp. 479–488.
[85] Y. Wexler, E. Shechtman, and M. Irani, *Space-Time Completion of Video*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (2007), pp. 463–476, doi:10.1109/TPAMI.2007.60.
[86] G. Xia, S. Ferradans, G. Peyré, and J. Aujol, *Synthesizing and Mixing Stationary Gaussian Texture Models*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 476–508, doi:10.1137/130918010.
[87] Z. Xu and J. Sun, *Image inpainting by patch propagation using patch sparsity*, IEEE Transactions on Image Processing, 19 (2010), pp. 1153–1165.