



HAL
open science

Texture Inpainting Using Efficient Gaussian Conditional Simulation

Bruno Galerne, Arthur Leclaire

► **To cite this version:**

Bruno Galerne, Arthur Leclaire. Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*, 2017, 10 (3), pp.1446-1474. 10.1137/16M1109047. hal-01428428v2

HAL Id: hal-01428428

<https://hal.science/hal-01428428v2>

Submitted on 29 Mar 2017

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

1 **TEXTURE INPAINTING USING EFFICIENT GAUSSIAN**
2 **CONDITIONAL SIMULATION**

3 BRUNO GALERNE* AND ARTHUR LECLAIRE†

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

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

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

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

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

32 General image inpainting is a very ill-posed problem, and instead of retrieving
33 the occluded content, one can only make a guess of what the image should have been.
34 However, in the restricted framework of textures, we have at our disposal several
35 stochastic models which can be used to model and synthesize a large class of textures.
36 In this setting, inpainting consists in first estimating a stochastic model from the
37 unmasked region, and then performing conditional simulation of the estimated random
38 model given the values around the mask. This point of view thus provides a better-
39 posed formulation of textural inpainting, which has been seldom considered in the
40 past. In particular, such approximate conditional sampling results are given in [31,
41 84, 54] under the name “constrained texture synthesis”. Also, the authors of [24] give
42 an instructive discussion which opposes deterministic and stochastic strategies for
43 image inpainting (with the intention to explain the differences between [31] and [84]).

44 It seems reasonable to assert that the choice between deterministic methods or
45 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.

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

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

62 **1.1. Inpainting Algorithms for Geometric Content.** As mentioned above,
 63 a very natural way to inpaint images is to propagate the geometric content through the
 64 masked region. To that purpose, the early geometric inpainting methods described
 65 by Masnou and Morel [68, 67] consist in connecting the level lines across the hole
 66 in order to satisfy the Gestaltist’s principle of good continuation. More precisely,
 67 the inpainted image is the solution of a generic minimization problem which includes
 68 the total variation (TV) of the image and the angle total variation of the level lines
 69 (Euler’s elastica).

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

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

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

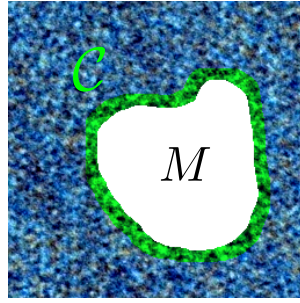


FIG. 1. *Textural inpainting via conditional simulation.* Inpainting with a stochastic texture model amounts to sampling the values on the mask M knowing the values on conditioning points C located at the border of the mask.

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

97 **1.2. Exemplar-based Inpainting, Sampling or Minimizing?...** An efficient
 98 way to model irregular images is to consider stochastic image models, and in partic-
 99 ular many texture synthesis algorithms can be formulated as sampling a probability
 100 distribution. Thus, one first strategy to inpaint textural parts of an image is to use
 101 an exemplar-based texture synthesis algorithm and to blend the synthesized content
 102 in the masked image. Such a method was proposed by Igehy and Pereira [49] who
 103 relied on Heeger-Bergen synthesis algorithm [47] to produce textural content.

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

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

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

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

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

162 **1.3. Gaussian Conditional Simulation.** In this paper, we will address textu-
 163 ral inpainting by precise conditional sampling of a stochastic texture model.

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

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

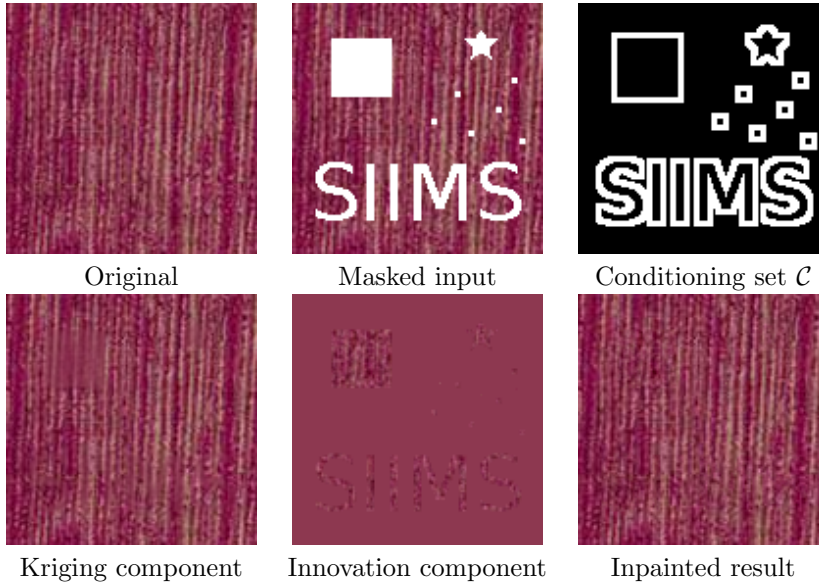


FIG. 2. **Summary of our microtexture inpainting method** The main idea of our method is to fill the masked region with a conditional sample of a Gaussian model. So this method is less about retrieving the initial image than computing another plausible sample of the texture model in the masked region. 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.

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

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

200 **1.4. Connections with geostatistics.** However, kriging-based Gaussian con-
 201 ditional simulation is a traditional method used for data interpolation in geostatistics
 202 [21, 55, 22, 43, 26]. Several parts of the method we propose are already well-known
 203 to geostatisticians, sometimes under other names. In particular, the ADSN model that
 204 we use is an instance of moving-average random fields [51, 70] whose spectral-based
 205 unconditional sampling algorithm is explained in [45, 20, 58]. The authors of [58]
 206 also suggest an optimization procedure to modify the unconditional sample so that
 207 it complies with the available data. In contrast, we propose direct sampling of a
 208 global conditional Gaussian model. Let us emphasize that, contrary to many exam-
 209 ples shown in the geostatistics literature, our imaging application leads to very large
 210 conditioning sets (with possibly several thousands conditioning values). Thus, in our
 211 case, precise conditional sampling is much more difficult than unconditional sampling.

212 Also, in the geostatistics literature, several authors have proposed generalized
 213 kriging algorithms for data prediction with various stochastic models [76, 1, 77, 33,
 214 21, 62]. In particular, in [76], Rue proposes a fast algorithm for conditional simulation
 215 in the particular case of Gaussian Markov random fields. Another technique for
 216 fast sampling in geostatistics is given by sequential simulation [42], which amounts
 217 to progressive filling of the pixels in a random order using successive conditional
 218 sampling. In our context, this approach would require to solve larger and larger kriging
 219 systems and would not be as efficient as our global approach. About progressive filling
 220 of the pixels, let us also mention a clear connection between the inpainting adaptation
 221 of [31] and the direct sampling method of [66]. We refer the interested reader to [65]
 222 for a much deeper discussion on the links between texture synthesis and multiple-point
 223 geostatistics.

224 **1.5. Plan of the Paper.** In Section 2, we explain the traditional algorithm for
 225 Gaussian conditional simulation (using a terminology that is derived from kriging es-
 226 timation). In Section 3, we apply this conditional sampling algorithm to microtexture
 227 inpainting. In particular, we discuss the estimation of a Gaussian model on a masked
 228 exemplar, and we also provide a Fourier based algorithm which allows to compute
 229 the kriging estimation even when the number of conditioning points is very large.
 230 Finally, in Section 4, we provide several texture inpainting experiments to illustrate
 231 the validity of our approach; in particular we show that our method can compete with
 232 state of the art inpainting methods on textural content.

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

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

242 We also introduce a subset $\mathcal{C} \subset \Omega$ of conditioning points. Given prescribed values
 243 $\varphi : \mathcal{C} \rightarrow \mathbb{R}$ on \mathcal{C} , conditional Gaussian simulation consists in sampling the conditional
 244 distribution of F given that $F|_{\mathcal{C}} = \varphi$. As we shall see later, this conditional sampling
 245 makes sense as soon as φ belongs to the support of the distribution of $F|_{\mathcal{C}}$, which is
 246 the range of the restricted covariance matrix $\Gamma|_{\mathcal{C} \times \mathcal{C}}$ and denoted by $\text{Range}(\Gamma|_{\mathcal{C} \times \mathcal{C}})$.

247 **2.1. Simple Kriging Estimation.** We define the **simple kriging estimator**

248 (1)
$$F^*(x) = \mathbb{E}(F(x) \mid F(c) , c \in \mathcal{C}).$$

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

253 (2)
$$F^*(x) = \sum_{c \in \mathcal{C}} \lambda_c(x) F(c).$$

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

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

259 **2.2. Gaussian Conditional Sampling Using Kriging Estimation.** Let us
 260 fix a set of coefficients $(\lambda_c(x))_{x \in \Omega, c \in \mathcal{C}}$ satisfying (2). For any $\varphi : \mathcal{C} \rightarrow \mathbb{R}$, we denote
 261 by φ^* the kriging estimation based on the values φ , defined for $x \in \Omega$ by $\varphi^*(x) =$
 262 $\sum_{c \in \mathcal{C}} \lambda_c(x) \varphi(c)$. With a notation abuse, if $\varphi : \Omega \rightarrow \mathbb{R}$, we will denote $\varphi^* = (\varphi|_{\mathcal{C}})^*$.

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

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

269 **2.3. Expression of the Kriging Coefficients.** In order to compute the kriging
 270 estimator at $x \in \Omega$, one needs to compute a valid set of kriging coefficients $(\lambda_c(x))_{c \in \mathcal{C}}$.
 271 Since F^* and $F - F^*$ are orthogonal, we get that the row vector $\lambda(x) = (\lambda_c(x))_{c \in \mathcal{C}}$ is
 272 a solution of the following $|\mathcal{C}| \times |\mathcal{C}|$ linear system

273 (3)
$$\forall c \in \mathcal{C}, \quad \sum_{d \in \mathcal{C}} \lambda_d(x) \Gamma(d, c) = \Gamma(x, c), \quad \text{i.e.} \quad \lambda(x) \Gamma|_{\mathcal{C} \times \mathcal{C}} = \Gamma_{|\{x\} \times \mathcal{C}}.$$

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

275 Aggregating the kriging coefficients in a $|\Omega| \times |\mathcal{C}|$ matrix $\Lambda = (\lambda_c(x))_{x \in \Omega, c \in \mathcal{C}}$, the
 276 system characterizing the kriging coefficients can also be written $\Lambda \Gamma|_{\mathcal{C} \times \mathcal{C}} = \Gamma_{|\Omega \times \mathcal{C}}$.
 277 If the matrix $\Gamma|_{\mathcal{C} \times \mathcal{C}}$ is invertible, the global system admits a unique solution $\Lambda =$
 278 $\Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^{-1}$. In the case where $\Gamma|_{\mathcal{C} \times \mathcal{C}}$ is not invertible, it is always possible to compute
 279 valid kriging coefficients with the pseudo-inverse $\Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger$. Indeed, since the system (3)
 280 has a solution¹, then $\Gamma_{|\{x\} \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger$ is also a solution. Thus we can always consider
 281 the set of kriging coefficients given by $\Lambda = \Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger$.

282 Once a set Λ of valid kriging coefficients has been computed, a conditional sample
 283 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
 284 column vectors.

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

285 **2.4. Matrix Expression of the Conditional Simulation.** From this expres-
 286 sion of the conditional sample, we will derive the usual expression of the Gaussian
 287 conditional distribution in matrix notation (as e.g. in [77, 74]).

288 Let $p = |\mathcal{C}|$, $q = |\Omega \setminus \mathcal{C}|$ (where $\Omega \setminus \mathcal{C}$ denotes the complement of \mathcal{C} in Ω) and
 289 $n = |\Omega|$. Let us introduce the matrices $R = \begin{pmatrix} I_p & 0 \end{pmatrix} \in \mathbb{R}^{p \times n}$, $S = \begin{pmatrix} 0 & I_q \end{pmatrix} \in \mathbb{R}^{q \times n}$,
 290 Using the first p indices for the elements of \mathcal{C} , we write block decompositions

$$291 \quad F = \begin{pmatrix} F|_{\mathcal{C}} \\ F|_{\Omega \setminus \mathcal{C}} \end{pmatrix} = \begin{pmatrix} RF \\ SF \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \Gamma|_{\mathcal{C} \times \mathcal{C}} & \Gamma|_{\mathcal{C} \times (\Omega \setminus \mathcal{C})} \\ \Gamma|_{(\Omega \setminus \mathcal{C}) \times \mathcal{C}} & \Gamma|_{(\Omega \setminus \mathcal{C}) \times (\Omega \setminus \mathcal{C})} \end{pmatrix} = \begin{pmatrix} R\Gamma R^T & R\Gamma S^T \\ S\Gamma R^T & S\Gamma S^T \end{pmatrix}.$$

292 With such notation, if $\varphi \in \text{Range}(\Gamma|_{\mathcal{C} \times \mathcal{C}})$, a conditional sample of F given $F|_{\mathcal{C}} = \varphi$ is
 293 given by $\Lambda\varphi + F - \Lambda R F$. From this expression we get the conditional distribution

$$294 \quad (4) \quad F | F|_{\mathcal{C}} = \varphi \sim \mathcal{N}(\Lambda\varphi, (I_n - \Lambda R)\Gamma(I_n - \Lambda R)^T).$$

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

$$296 \quad (5) \quad \mathbb{E}(SF | F|_{\mathcal{C}} = \varphi) = S\Lambda\varphi = S \begin{pmatrix} R\Gamma R^T \\ S\Gamma R^T \end{pmatrix} (R\Gamma R^T)^\dagger \varphi = S\Gamma R^T (R\Gamma R^T)^\dagger \varphi,$$

$$297 \quad (6) \quad \text{Cov}(SF | F|_{\mathcal{C}} = \varphi) = S\Gamma S^T - S\Gamma R^T (R\Gamma R^T)^\dagger R\Gamma S^T.$$

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

300 **3. Microtexture Inpainting Algorithm.** This section contains our main con-
 301 tribution: how to use Gaussian conditional sampling for microtexture inpainting.

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

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

315 **3.1. ADSN Models.** As shown in [83, 36], a convenient model for microtexture
 316 is given by the asymptotic discrete spot noise (ADSN). Given a function $h : \mathbb{Z}^2 \rightarrow \mathbb{R}$
 317 with finite support, the ADSN corresponding to h is the convolution of h with a
 318 normalized Gaussian white noise W on \mathbb{Z}^2 , defined as

$$319 \quad (7) \quad \forall x \in \mathbb{Z}^2, \quad h * W(x) = \sum_{y \in \mathbb{Z}^2} h(y)W(x - y).$$

320 This Gaussian random field is stationary, has zero mean, and its covariance function
 321 is given by $\mathbb{E}(h * W(x)h * W(y)) = (h * \tilde{h})(x - y)$, where $\tilde{h}(z) = h(-z)$. The restriction
 322 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
 323 simple convolutive expression of the ADSN, it can be efficiently sampled using the fast
 324 Fourier transform (FFT). Depending on the boundary conditions, we can consider a

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

330 **Extension to Color Images.** ADSN models extend to color images by con-
 331 volving each color channel with the same white noise in (7). This gives an \mathbb{R}^d -valued
 332 Gaussian random field F on Ω (where d is the number of channels, i.e. 3 for color
 333 images). Regarding the conditional simulation, a simple way to understand this ex-
 334 tension is to consider the \mathbb{R}^d -valued random field F as a real-valued random field on
 335 $\Omega \times \{1, \dots, d\}$. The covariance matrix is then given by

$$336 \quad (8) \quad \forall (x, j), (y, k) \in \Omega \times \{1, \dots, d\}, \quad \Gamma((x, j), (y, k)) = \mathbb{E}(F_j(x)F_k(y)).$$

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

340 **3.2. Estimation of the Gaussian Model.** If the image $u : \Omega \rightarrow \mathbb{R}^d$ were
 341 entirely available, the estimation procedure would be the same as for texture syn-
 342 thesis [36, 38], which is briefly recalled here. We compute the mean value $\bar{u} =$
 343 $\frac{1}{|\Omega|} \sum_{x \in \Omega} u(x)$ and the normalized spot $t_u = \frac{1}{\sqrt{|\Omega|}}(u - \bar{u})$ (extended by zero-padding).
 344 The microtexture u is then synthesized by sampling $\bar{u} + t_u * W$, with W a normalized
 345 Gaussian white noise. We call *oracle model* the ADSN model estimated from the
 346 unmasked exemplar.

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

$$350 \quad (9) \quad \bar{v} = \frac{1}{|\omega|} \sum_{x \in \omega} v(x), \quad t_v(x) = \begin{cases} \frac{1}{\sqrt{|\omega|}}(v(x) - \bar{v}) & \text{if } x \in \omega, \\ 0 & \text{otherwise.} \end{cases}$$

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

$$352 \quad (10) \quad c_v(h) = \frac{1}{|\omega|} \sum_{x \in \omega \cap (\omega - h)} (u(x+h) - \bar{v})(u(x) - \bar{v})^T \in \mathbb{R}^{d \times d}.$$

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

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

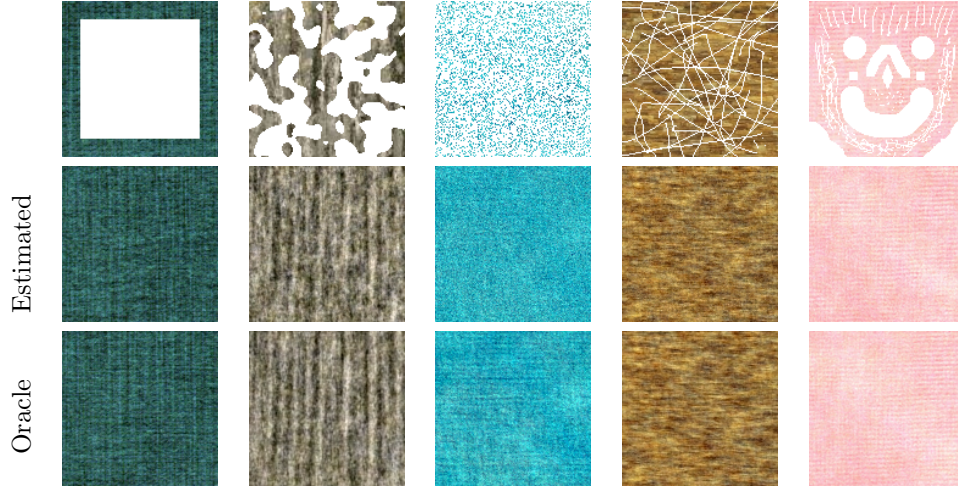


FIG. 3. *Estimation of an ADSN model on a masked exemplar.* We illustrate with several types of mask the estimation of the Gaussian model with the naive estimator (10) using $\omega = \Omega \setminus M$. We display in the first row the masked exemplar, in the second row a sample of the estimated ADSN 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).

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

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

376 **3.3. Kriging Estimation with Conjugate Gradient Descent.** In this sec-
 377 tion, we propose an efficient way to compute a conditional sample of the ADSN model.
 378 The most difficult part consists in solving a large linear system involving the condi-
 379 tional values. This step is dealt with by using a conjugate gradient descent algorithm,
 380 which proves to be efficient even for very large images.

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

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

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

391 (11)
$$(u - \bar{v})^* + F - F^* = \Lambda((u - \bar{v})|_{\mathcal{C}}) + F - \Lambda(F|_{\mathcal{C}}).$$

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

393 Let us begin with the multiplication by $\Gamma_{\Omega \times \mathcal{C}}$, which is easier. Assume that
 394 $\psi = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger \varphi$ has been computed. Using the notation of Section 2.4, $\Gamma_{|\Omega \times \mathcal{C}} \psi = \Gamma \Psi$,
 395 where $\Psi = R^T \psi \in \mathbb{R}^\Omega$ is the zero-padding extension of ψ . Now, since Γ is the
 396 covariance function of an ADSN model, it can be simply computed by convolution.
 397 More precisely, $\Gamma \Psi$ is the restriction on Ω of the convolution of Ψ by $t_v * \tilde{t}_v$.

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

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

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

412 (12)
$$f : \psi \mapsto \frac{1}{2} \|A\psi - \varphi\|^2$$

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

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

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

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

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

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

430 **Stopping criterion.** The stopping criterion that we use in Algorithm CGD is
 431 $\|r_k\| \leq \varepsilon$ where the residual at iteration k is given by

$$432 \quad (13) \quad r_k = A^T \varphi - A^T A \psi_k,$$

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

437 3.4. Comments on the Kriging System.

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

$$441 \quad (14) \quad \widehat{t}_v(\xi) \widehat{t}_v(\xi)^* = \begin{cases} \frac{1}{|\omega|} \widehat{v}(\xi) \widehat{v}(\xi)^* & \text{if } \xi \neq 0 \\ 0 & \text{if } \xi = 0 \end{cases}.$$

442 As noted in [86], this matrix has rank ≤ 1 which constrains the rank of the ma-
 443 trix Γ (of size $d|\Omega| \times d|\Omega|$) to be bounded by $|\Omega| - 1$. Since A is a submatrix of Ω ,
 444 $\text{Rank}(A) \leq |\Omega| - 1$. In particular, if the conditioning set is sufficiently big so that
 445 $d|\mathcal{C}| \geq |\Omega|$, then A cannot be invertible.

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

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

454 **3.5. Complete Algorithm.** To end this section, we summarize our microtex-
 455 ture inpainting algorithm. In Algorithm CGD the matrix $A = \Gamma|_{\mathcal{C} \times \mathcal{C}}$ is not formed
 456 explicitly, and we only need to apply it efficiently with the FFT-based algorithm.
 457 Also, if one is not interested in the kriging and innovation components but only in
 458 the inpainting result, then only one instance of gradient descent is needed since the
 459 output only depends on $(u - \bar{v} - F)^* = \Gamma|_{\mathcal{C} \times \mathcal{C}}^\dagger (u|_{\mathcal{C}} - \bar{v} - F|_{\mathcal{C}})$.

460 The overall complexity of this algorithm is $\mathcal{O}(k_{\max} |\Omega| \log |\Omega|)$ where k_{\max} is the
 461 number of iterations used in the gradient descent algorithm. The overall number of
 462 FFTs required by the whole inpainting process (whose detailed computation can be
 463 found in the IPOL companion paper) is $(4k_{\max} + 6)d$ FFTs. Using our C implemen-
 464 tation (involving parallel computing, in particular for the FFT) run with a modern
 465 computer (Intel i7 processor @2.60GHz with 4 cores), the whole inpainting process
 466 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

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

- Draw a Gaussian sample $F = t_v * W$
- Compute $\psi_1 = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger (u|_{\mathcal{C}} - \bar{v})$, $\psi_2 = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger F|_{\mathcal{C}}$
(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 $\bar{v} + (u - \bar{v})^* + F - F^*$

467

4. Results and Discussion.

468

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.

477

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.

489

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).

494

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

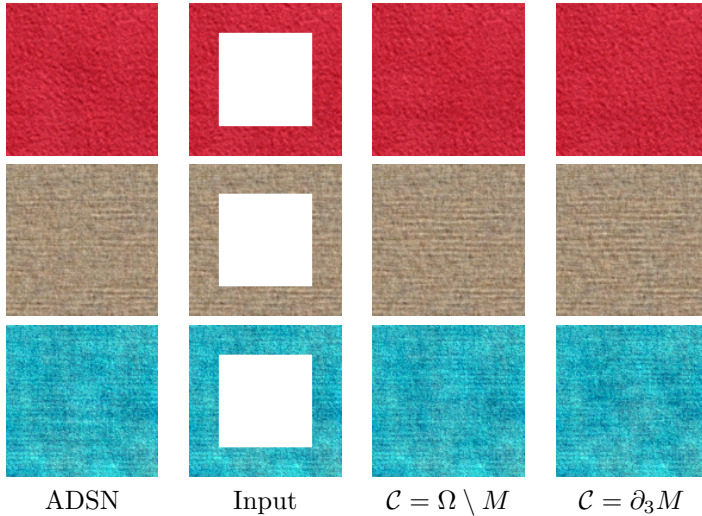


FIG. 4. *Inpainting Gaussian textures with the oracle Gaussian model - regular masks.* The masked input has been inpainted with Gaussian conditional simulation using an oracle Gaussian model (estimated from the unmasked exemplar texture) based on conditioning values on $\mathcal{C} \subset \Omega$. From 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.

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

499 **4.2. Inpainting with an Estimated Gaussian Model.** In this section, we
 500 provide experimental results which show that our algorithm is able to inpaint holes
 501 in microtextures, whatever the size of the hole, and with only minimal requirements
 502 on the hole regularity. In contrast with the last section, the Gaussian model is now
 503 estimated from the masked exemplar. We will show that the naive estimation tech-
 504 nique explained in Section 3.2 and illustrated in Fig. 3 leads to satisfying inpainting
 505 results except in the case where the mask is made of randomly scattered pixels. In
 506 the experiments shown in this section, we took $\mathcal{C} = \partial_3 M$.

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

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

518 **4.3. Computing and Visualizing the Kriging Coefficients.** In order to
 519 better understand the conditional simulation, it is interesting to visualize the kriging

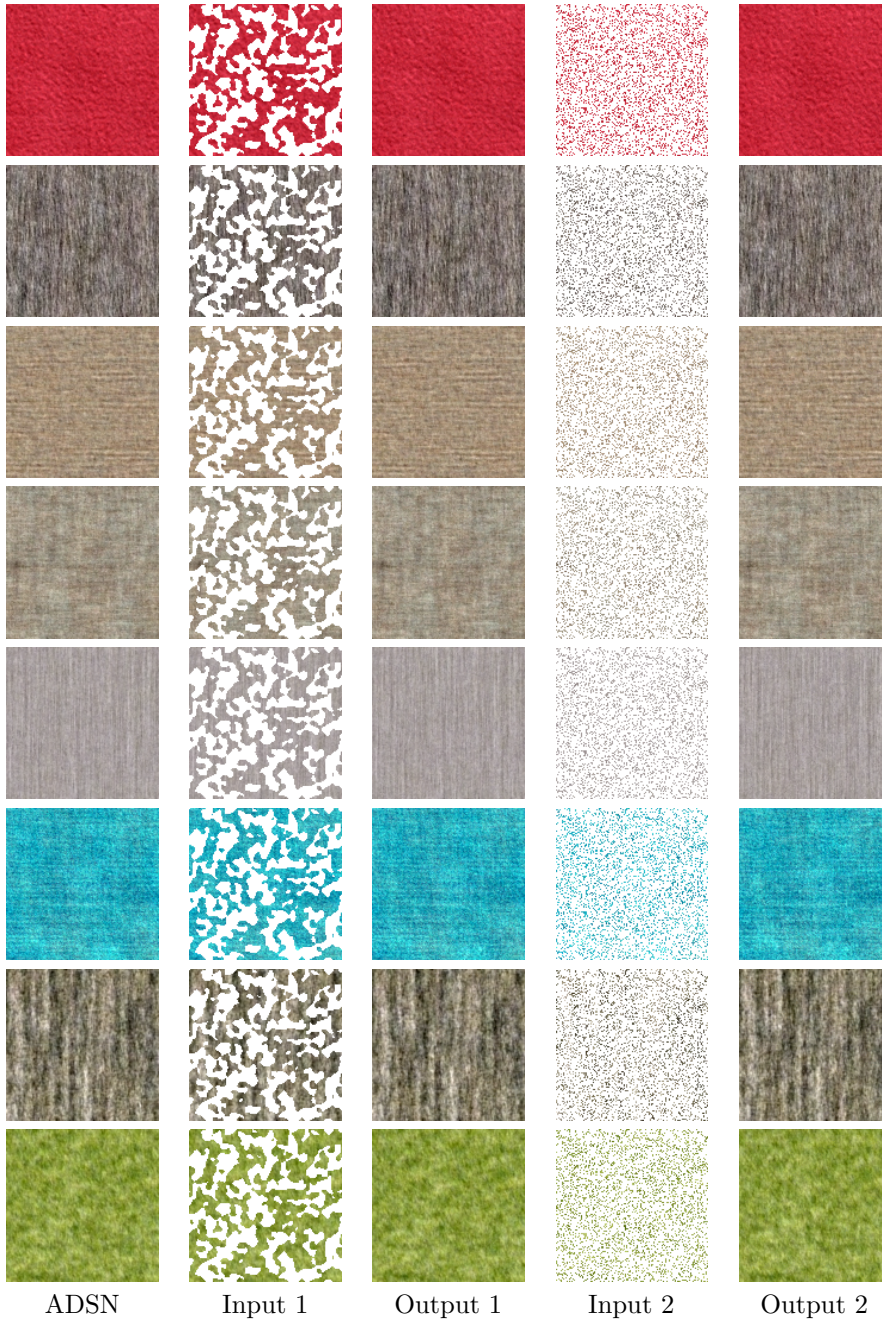


FIG. 5. *Inpainting Gaussian textures with the oracle Gaussian model - irregular masks.* The masked input has been inpainted with Gaussian conditional simulation using an oracle Gaussian model (estimated from the unmasked exemplar texture) based on conditioning values on $C \subset \Omega$. From left 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).

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

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

$$525 \quad (15) \quad (\lambda_c(x))_{x \in \Omega} = \Lambda \delta_c = \Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger \delta_c,$$

526 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
 527 use our algorithm on a Dirac input.

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

$$530 \quad (16) \quad (\lambda_c(x))_{c \in \mathcal{C}} = \Lambda^T \delta_x = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger \Gamma_{|\mathcal{C} \times \Omega} \delta_x,$$

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

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

538 **4.4. Impact of the Size of the Conditioning Border.** In this section, we
 539 investigate the impact of changing the size of the conditioning border. Again, an ideal
 540 setting would be to choose $\mathcal{C} = \Omega \setminus M$, but then the kriging system is very large. Here
 541 we will confirm that taking $\mathcal{C} = \partial_w M$ is sufficient, and we will precisely examine the
 542 variation of the conditional model when increasing the width w of the border.

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

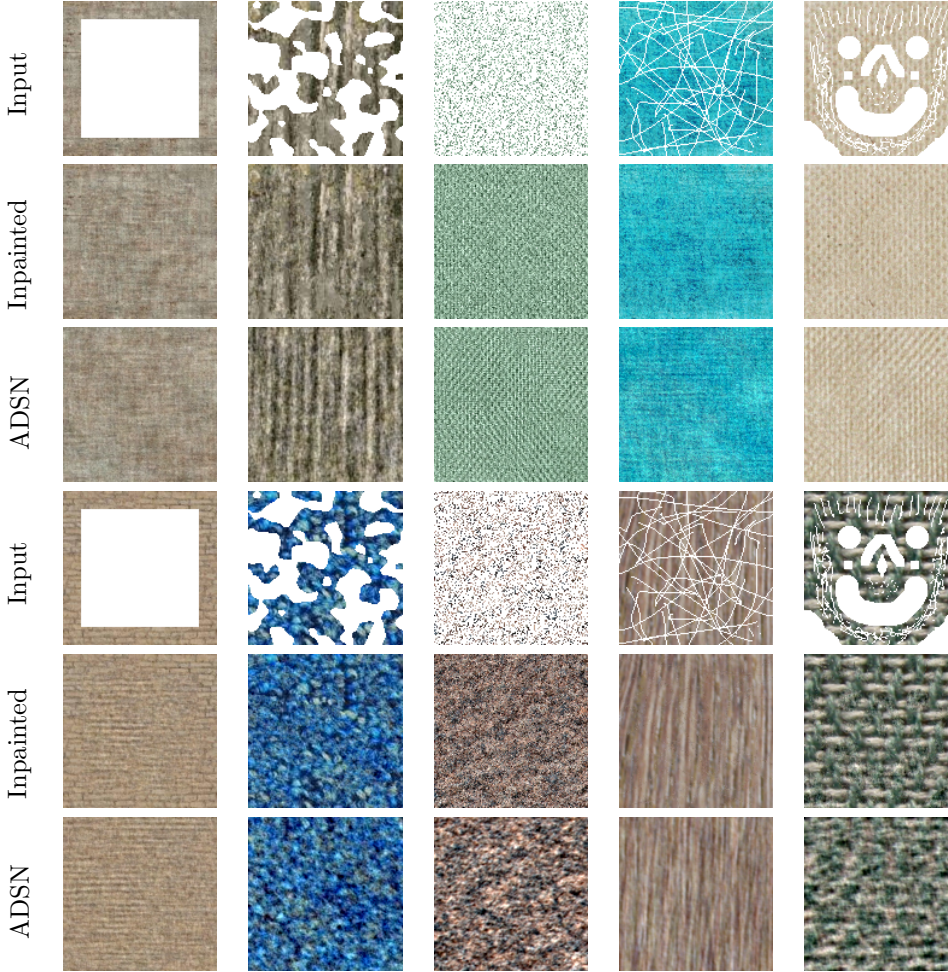


FIG. 7. **Examples of textural inpainting.** We present results of our inpainting method for several textures and masks. From top to bottom (rows 1-3 and rows 4-6), we display a masked input, the inpainted result, and a sample of the estimated ADSN model (which is useful to exhibit the limit of the Gaussian 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.

545 A possible way to perform this comparison is to rely on the L^2 -optimal transport
 546 distance, which has already been used in several works about texture synthesis [86, 38].
 547 Let us recall [29] that the L^2 -optimal transport distance between two Gaussian models
 548 $\mu_X = \mathcal{N}(m_X, \Sigma_X), \mu_Y = \mathcal{N}(m_Y, \Sigma_Y)$ is given by

$$549 \quad (17) \quad d_{\text{OT}}(\mu_X, \mu_Y)^2 = \|m_X - m_Y\|^2 + \text{Tr}(\Sigma_X) + \text{Tr}(\Sigma_Y) - 2\text{Tr}\left((\Sigma_X \Sigma_Y)^{1/2}\right).$$

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

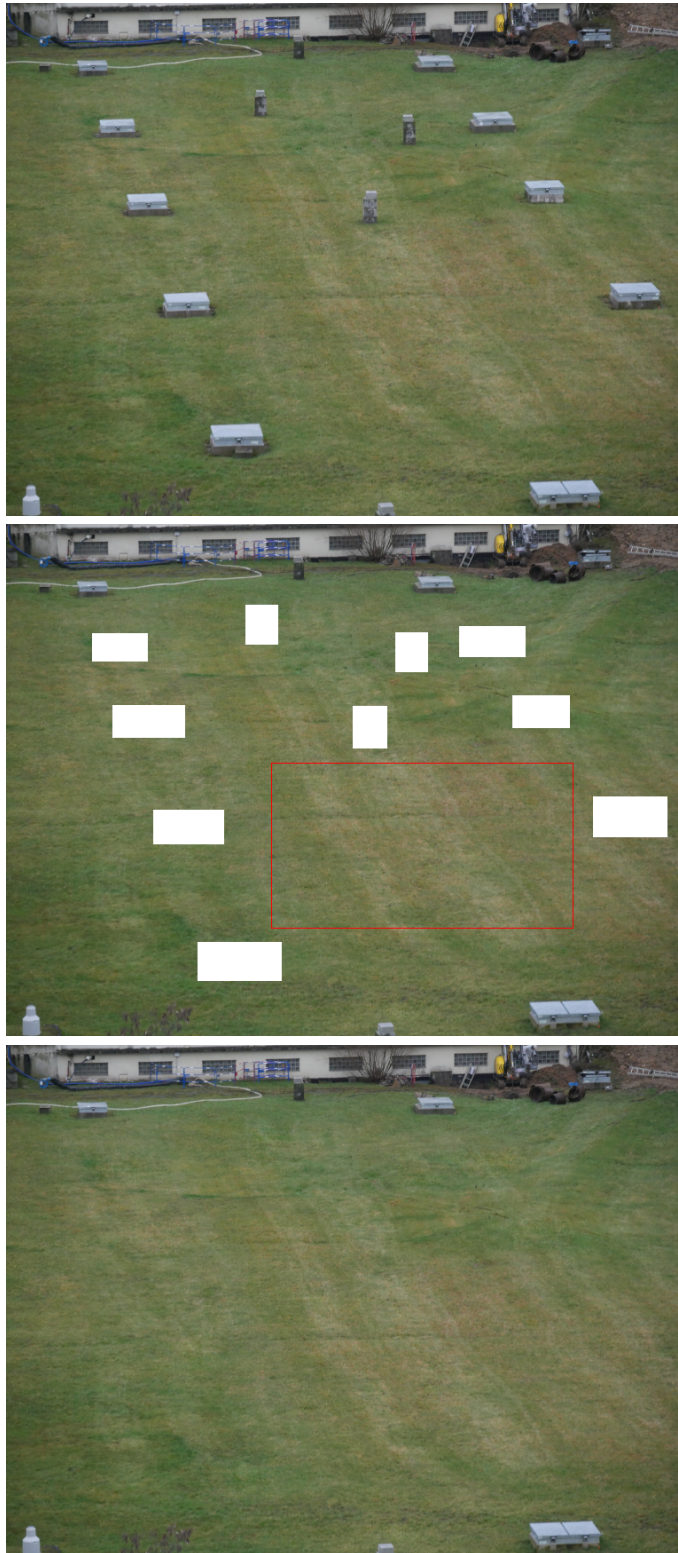


FIG. 8. *Inpainting textural parts of an image.* From top to bottom, we display the original image (of size 768×577), the masked input (the Gaussian model has been estimated in the subdomain ω delimited by the red box), and the inpainted result. Our algorithm is able to synthesize microtexture content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

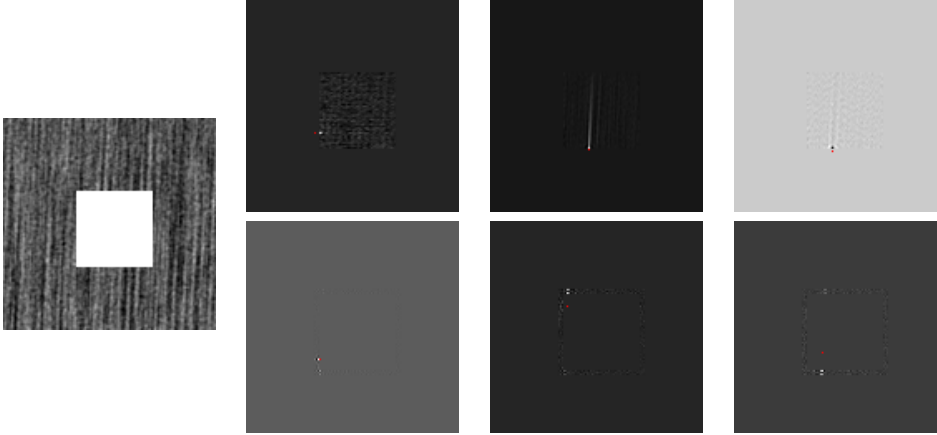


FIG. 9. **Visualizing Kriging coefficients.** In the first column, we display the masked input. For the three other columns: in the first row, we display the kriging coefficients $(\lambda_c(x))_{x \in M}$ for different positions of the conditioning pixel $c \in \mathcal{C}$ (drawn in red); in the second 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.

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

$$555 \quad m_w = \Gamma_{|M \times \mathcal{C}_w} \Gamma_{|\mathcal{C}_w \times \mathcal{C}_w}^\dagger (u - \bar{u})_{|\mathcal{C}_w}, \quad \Sigma_w = \Gamma_{|M \times M} - \Gamma_{|M \times \mathcal{C}_w} \Gamma_{|\mathcal{C}_w \times \mathcal{C}_w}^\dagger \Gamma_{|\mathcal{C}_w \times M}.$$

556 For our experiment, we choose a reasonably small texture so that all these covariance
557 matrices can be explicitly built and stored (relying on standard numerical routines
558 for pseudo-inverse and square roots computation²). We then plot the function

$$559 \quad (18) \quad w \in \{1, \dots, 20\} \mapsto \frac{d_{\text{OT}}(\mu_w, \mu_\infty)}{\sigma_u \sqrt{|M|}},$$

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

$$562 \quad d(m_w, m_\infty) = \|m_w - m_\infty\|, \quad d(\Sigma_w, \Sigma_\infty)^2 = \text{Tr}(\Sigma_w) + \text{Tr}(\Sigma_\infty) - 2\text{Tr}((\Sigma_w \Sigma_\infty)^{1/2}).$$

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

²The 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} .

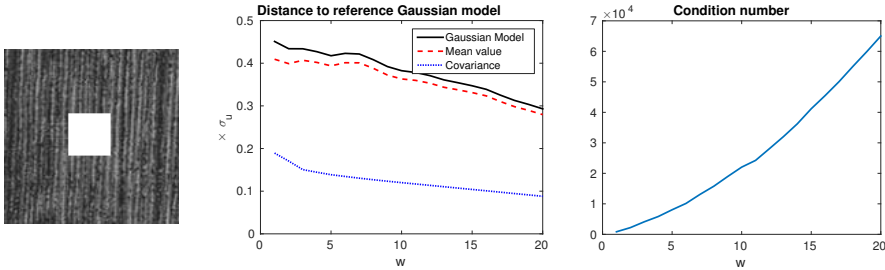


FIG. 10. *Quantitative study of the conditional models depending on the conditioning set \mathcal{C} . We computed the distance between the reference conditional model (obtained for $\mathcal{C}_\infty = \Omega \setminus M$) and the conditional models (obtained for $\mathcal{C}_w = \partial_w M$), see (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.*

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

577 It is interesting to notice that the kriging components look very different with
 578 $w = 5$ and $w = \infty$. Indeed, when the conditioning set gets larger, the kriging com-
 579 ponent depends on a larger number of random variables, and thus has an increased
 580 stochastic nature. This explains why the distance between the Gaussian models (or
 581 their mean or covariance functions) does not quickly tend to zero when w increases.
 582 Still, as reflected by the example of Fig. 11 and as observed in all our experiments,
 583 the inpainting result is already good for $w = 3$ (in particular, for many textures, this
 584 value is sufficient to naturally blend the inpainted domain in the context).

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

588 **4.5. Comparisons.** In this section, we compare our microtexture inpainting
 589 algorithm with several recent inpainting techniques.

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

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

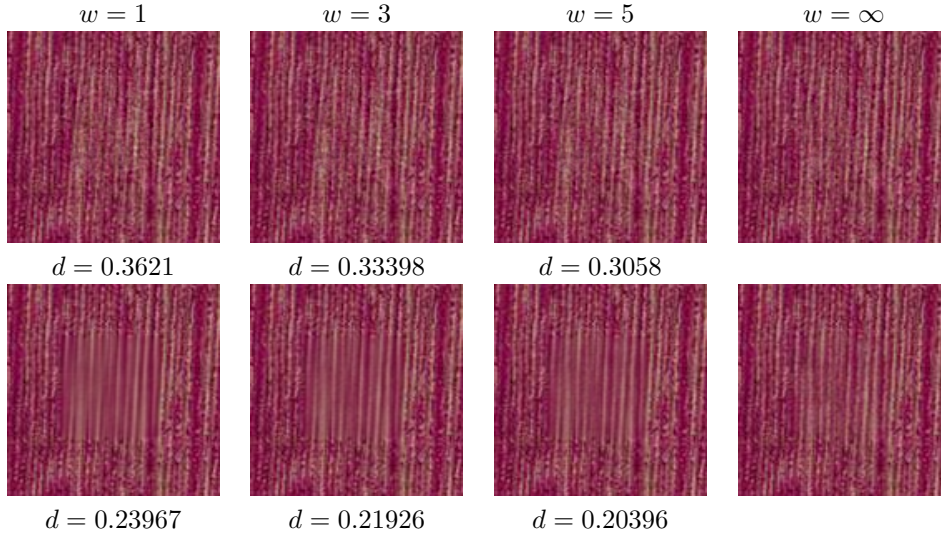


FIG. 11. *Qualitative study of the conditional models depending on the conditioning set C . From left to right, we display the inpainting results obtained for C being a border of M of width $w = 1, 3, 5$ pixels, and also the limit solution $C = \Omega \setminus M$. In the first row, we display the sample 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.*

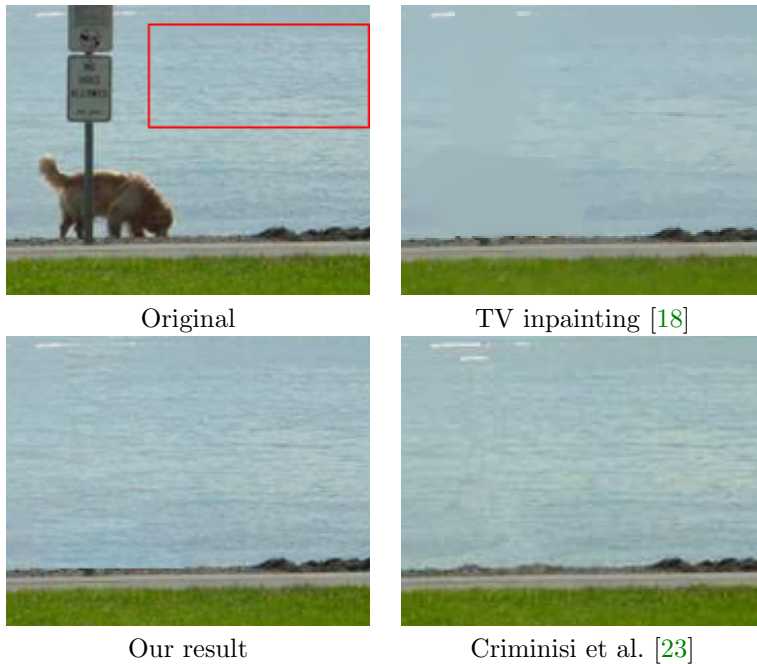


FIG. 12. *Comparison with [18, 23]. In the first row, we display the original image (taken from [23]) on the left, and on the right the result of TV inpainting [18] (obtained with the implementation available at [41]). In the second row, on the left we show the result of Gaussian inpainting (with a model 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.*

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

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

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

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

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

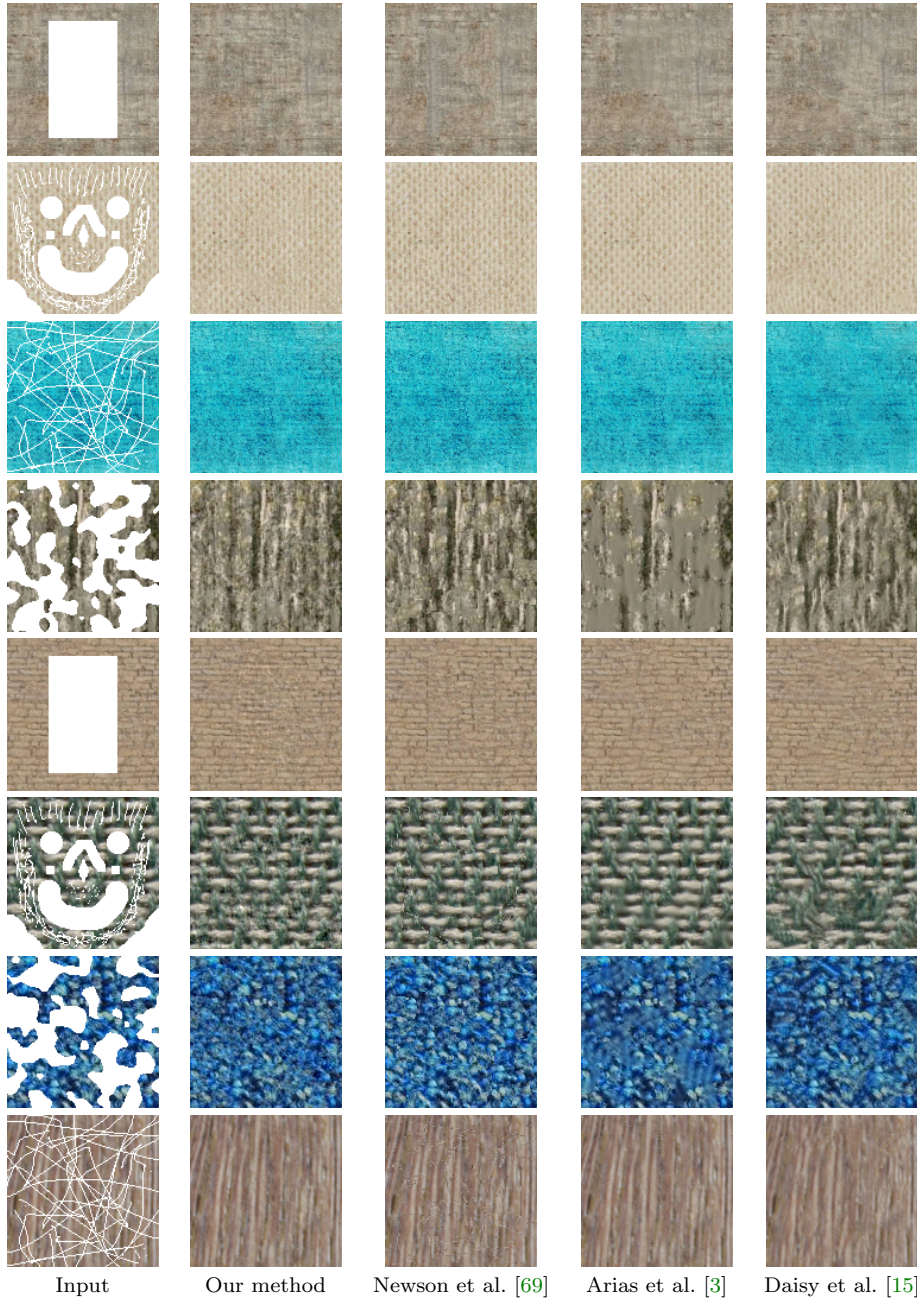


FIG. 13. *Comparison with patch-based methods (I).* On each row, from left to right, we display a masked input, the result of our Gaussian inpainting algorithm, the result of [69], the result of variational non-local inpainting [3] (obtained with the online implementation of [35] using the NLmeans option), 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.

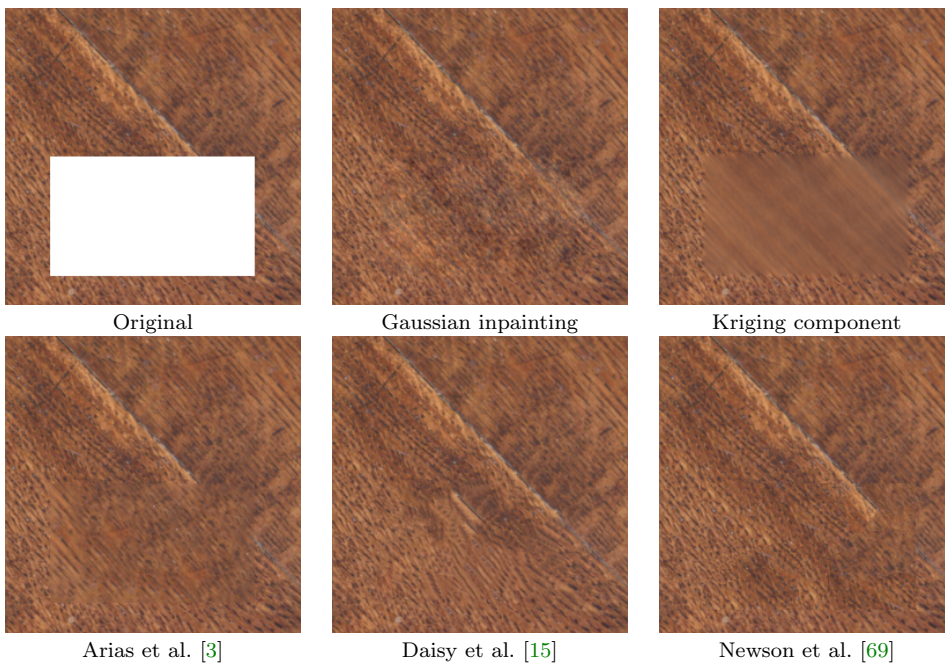


FIG. 14. *Comparison with patch-based methods (II)*. We compare several inpainting methods on a difficult textural inpainting problem. On the first row, from left or right, we display the masked input, the result of our method, together with the corresponding kriging component. On the second row, we 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.

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

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

REFERENCES

- 657
- 658 [1] A. ALMANSA, F. CAO, Y. GOUSSEAU, AND B. ROUGÉ, *Interpolation of digital elevation models*
 659 *using AMLE and related methods*, IEEE transactions on geoscience and remote sensing,
 660 40 (2002), pp. 314–325.
- 661 [2] P. ARIAS, V. CASELLES, AND G. FACCIOLO, *Analysis of a Variational Framework for Exemplar-*
 662 *Based Image Inpainting*, Multiscale Modeling & Simulation, 10 (2012), pp. 473–514, doi:10.
 663 1137/110848281.
- 664 [3] P. ARIAS, G. FACCIOLO, V. CASELLES, AND G. SAPIRO, *A variational framework for exemplar-*
 665 *based image inpainting*, International Journal of Computer Vision, 93 (2011), pp. 319–347.
- 666 [4] J. AUJOL, S. LADJAL, AND S. MASNOU, *Exemplar-based inpainting from a variational point of*
 667 *view*, SIAM Journal on Mathematical Analysis, 42 (2010), pp. 1246–1285.
- 668 [5] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO, AND J. VERDERA, *Filling-in by joint*

- 669 *interpolation of vector fields and graylevels*, IEEE Transactions on Image Processing, 10
 670 (2001), pp. 1200–1211.
- 671 [6] C. BARNES, E. SHECHTMAN, A. FINKELSTEIN, AND D. GOLDMAN, *PatchMatch: a randomized*
 672 *correspondence algorithm for structural image editing*, ACM Transactions on Graphics, 28
 673 (2009).
- 674 [7] M. BERTALMIO, A. BERTOZZI, AND G. SAPIRO, *Navier-stokes, fluid dynamics, and image and*
 675 *video inpainting*, in Proceedings of CVPR, vol. 1, IEEE, 2001.
- 676 [8] M. BERTALMIO, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image Inpainting*, in Proc. of
 677 SIGGRAPH, 2000, pp. 417–424, doi:10.1145/344779.344972.
- 678 [9] M. BERTALMIO, L. VESE, G. SAPIRO, AND S. OSHER, *Simultaneous structure and texture image*
 679 *inpainting*, IEEE Transactions on Image Processing, 12 (2003), pp. 882–889.
- 680 [10] A. BERTOZZI, S. ESEDOGLU, AND A. GILLETTE, *Inpainting of binary images using the Cahn-*
 681 *Hilliard equation*, IEEE Transactions on image processing, 16 (2007), pp. 285–291.
- 682 [11] R. BORNARD, E. LECAN, L. LABORELLI, AND J. CHENOT, *Missing data correction in still im-*
 683 *ages and image sequences*, in Proceedings of the tenth ACM international conference on
 684 Multimedia, 2002, pp. 355–361.
- 685 [12] F. BORNEMANN AND T. MÄRZ, *Fast image inpainting based on coherence transport*, Journal of
 686 Mathematical Imaging and Vision, 28 (2007), pp. 259–278.
- 687 [13] A. BUGEAU, M. BERTALMO, V. CASELLES, AND G. SAPIRO, *A comprehensive framework for*
 688 *image inpainting*, IEEE Transactions on Image Processing, 19 (2010), pp. 2634–2645.
- 689 [14] M. BURGER, L. HE, AND C. SCHÖNLIEB, *Cahn-Hilliard inpainting and a generalization for*
 690 *grayvalue images*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 1129–1167.
- 691 [15] P. BUYSSENS, M. DAISY, D. TSCHUMPERLÉ, AND O. LÉZORAY, *Exemplar-based Inpainting:*
 692 *Technical Review and new Heuristics for better Geometric Reconstructions*, IEEE Trans-
 693 actions on Image Processing, 24 (2015), pp. 1809–1824.
- 694 [16] J. CAI, R. CHAN, AND Z. SHEN, *A framelet-based image inpainting algorithm*, Applied and
 695 Computational Harmonic Analysis, 24 (2008), pp. 131–149.
- 696 [17] F. CAO, Y. GOUSSEAU, S. MASNOU, AND P. PÉREZ, *Geometrically guided exemplar-based in-*
 697 *painting*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 1143–1179.
- 698 [18] T. CHAN AND J. SHEN, *Mathematical models for local nontexture inpaintings*, SIAM Journal
 699 on Applied Mathematics, 62 (2002), pp. 1019–1043.
- 700 [19] S. CHANDRA, M. PETROU, AND R. PIRODDI, *Texture Interpolation Using Ordinary Kriging*,
 701 Pattern Recognition and Image Analysis, (2005), pp. 183–190.
- 702 [20] J. CHILÈS, *Quelques méthodes de simulation de fonctions aléatoires intrinsèques*, Cahiers de
 703 Géostatistique, 5 (1995), pp. 97–112.
- 704 [21] J.-P. CHILÈS AND P. DELFINER, *Geostatistics: modeling spatial uncertainty*, John Wiley &
 705 Sons, 2009.
- 706 [22] N. CRESSIE, *Statistics for Spatial Data*, John Wiley & Sons, 1993.
- 707 [23] A. CRIMINISI, P. PÉREZ, AND K. TOYAMA, *Region filling and object removal by exemplar-based*
 708 *image inpainting*, IEEE Transactions on Image Processing, 13 (2004), pp. 1200–1212.
- 709 [24] L. DEMANET, B. SONG, AND T. CHAN, *Image inpainting by correspondence maps: a determin-*
 710 *istic approach*, Applied and Computational Mathematics, 1100 (2003), p. 99.
- 711 [25] A. DESOLNEUX, L. MOISAN, AND S. RONSIN, *A compact representation of random phase and*
 712 *Gaussian textures*, in Proceedings of ICASSP, 2012, pp. 1381–1384.
- 713 [26] C. DEUTSCH AND A. JOURNEL, *Geostatistical software library and user’s guide*, vol. 119, Oxford
 714 University Press, New York, 1992.
- 715 [27] P. DIGGLE, J. TAWN, AND R. MOYEEED, *Model-based geostatistics*, Journal of the Royal Statis-
 716 tical Society: Series C (Applied Statistics), 47 (1998), pp. 299–325.
- 717 [28] J. DOOB, *Stochastic processes*, Wiley, 1990.
- 718 [29] D. C. DOWSON AND B. V. LANDAU, *The Fréchet distance between multivariate normal distribu-*
 719 *tions*, Journal of Multivariate Analysis, 12 (1982), pp. 450–455, doi:10.1016/0047-259X(82)
 720 90077-X.
- 721 [30] I. DRORI, D. COHEN-OR, AND H. YESHURUN, *Fragment-based image completion*, in ACM Trans-
 722 actions on Graphics, vol. 22, 2003, pp. 303–312.
- 723 [31] A. A. EFROS AND T. K. LEUNG, *Texture synthesis by non-parametric sampling*, in Proceedings
 724 of ICCV, vol. 2, 1999, pp. 1033–1038.
- 725 [32] M. ELAD, J. L. STARCK, P. QUERRE, AND D. L. DONOHO, *Simultaneous cartoon and texture*
 726 *image inpainting using morphological component analysis (MCA)*, Applied and Computa-
 727 tional Harmonic Analysis, 19 (2005), pp. 340–358, doi:10.1016/j.acha.2005.03.005.
- 728 [33] X. EMERY, *Conditioning Simulations of Gaussian Random Fields by Ordinary Kriging*, Math-
 729 ematical Geology, 39 (2007), pp. 607–623, doi:10.1007/s11004-007-9112-x.
- 730 [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-HERNÁNDEZ 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 IICIP, 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 priori-*
775 *ty 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. LANTUÉJOL, *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. MARIETHOZ 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. MARIETHOZ 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.