



HAL
open science

Gaussian Texture Inpainting

Bruno Galerne, Arthur Leclaire

► **To cite this version:**

| Bruno Galerne, Arthur Leclaire. Gaussian Texture Inpainting. 2016. hal-01428428v1

HAL Id: hal-01428428

<https://hal.science/hal-01428428v1>

Preprint submitted on 6 Jan 2017 (v1), last revised 29 Mar 2017 (v2)

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

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

GAUSSIAN TEXTURE INPAINTING

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 [54, 53]. Both these works suggest to fill the hole by extending the geometric structures, either by level-lines completion [54] 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 [27, 67] under the name “constrained texture synthesis”. Also, the authors of [22] give an instructive discussion which opposes deterministic and stochastic strategies for image inpainting (with the intention to explain the differences between [27] and [67]).

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 will mainly focus on inpainting very irregular texture images, called microtextures.

*Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité, France.

†CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France.

46 These textures are not well described by a generic variational principle. In contrast,
 47 they can be precisely and efficiently synthesized with simple stochastic models that
 48 rely on second-order statistics, for example the asymptotic discrete spot noise (ADSN)
 49 introduced in [66] and thoroughly studied in [31, 69, 46]. In this paper, we propose a
 50 microtexture inpainting algorithm that relies on a precise conditional sampling. Con-
 51 ditional sampling of the ADSN model can be easily formulated, and gives inpainting
 52 results which are visually better than the ones obtained with recent methods while
 53 keeping strong mathematical guarantees.

54 In the remaining paragraphs of this introduction, we discuss existing inpainting
 55 techniques, and in particular discuss the links between image inpainting and texture
 56 synthesis. Giving an exhaustive overview of the literature on this famous problem is
 57 not the main purpose of this paper. We refer the interested reader to [36, 15, 62] for
 58 much more detailed reviews of existing methods.

59 **1.1. Inpainting Algorithms for Geometric Content.** As mentioned above,
 60 a very natural way to inpaint images is to propagate the geometric content through the
 61 masked region. To that purpose, the early geometric inpainting methods described
 62 by Masnou and Morel [54, 53] consist in connecting the level lines across the hole
 63 in order to satisfy the Gestaltist’s principle of good continuation. More precisely,
 64 the inpainted image is the solution of a generic minimization problem which includes
 65 the total variation (TV) of the image and the angle total variation of the level lines
 66 (Euler’s elastica).

67 Closely related to these generic variational inpainting methods lie models based
 68 on partial differential equations (PDE). Bertalmio et al. [8] suggest to iterate a finite-
 69 difference scheme, which was later interpreted as a numerical scheme for a PDE
 70 related to Navier-Stokes equation [7]. Of course, there is a strong connection between
 71 PDE-based and variational methods because the minimum of a generic functional
 72 satisfies the associated Euler-Lagrange equation (but a PDE may not be associated
 73 to a variational problem [62]). Among many papers lying in between PDEs and
 74 generic variational problems, we will only quote a few important contributions.

75 Ballester et al. [5] propose to perform joint interpolation of image values and gra-
 76 dient orientations by solving a minimization problem which leads to coupled second-
 77 order PDEs on image values and gradient orientations. Chan and Shen [18] give a
 78 detailed study of the inpainting method based on TV minimization (which, compared
 79 to [54] drops the elastica term in the minimization problem), and propose a more
 80 general scheme called curvature-driven diffusion (which allows to better respect the
 81 good continuation principle). The link with Mumford-Shah image model was already
 82 discussed in [18], and more importantly exploited by Esedoglu and Shen [29], who
 83 completed the Mumford-Shah model with an Euler’s elastica term, leading to fourth-
 84 order nonlinear parabolic PDEs, and allowing better connectivity in the inpainting
 85 result. Later, other fourth-order PDEs were exploited to inpaint non-texture images
 86 with better connectivity: Bertozzi et al. [10] propose to solve a modified Cahn-Hilliard
 87 equation for fast inpainting of binary or highly-contrasted images, an approach which
 88 was generalized to real-valued images by Burger et al. [14]. Finally, Bornemann and
 89 März [12] propose an efficient non-iterative inpainting algorithm which is based on a
 90 transport equation and inspired by the fast marching algorithm of [64].

91 A common drawback of these deterministic methods is that they are not able
 92 to inpaint textural content precisely because solving a PDE or a variational problem
 93 often imposes a certain degree of smoothness for the solution.

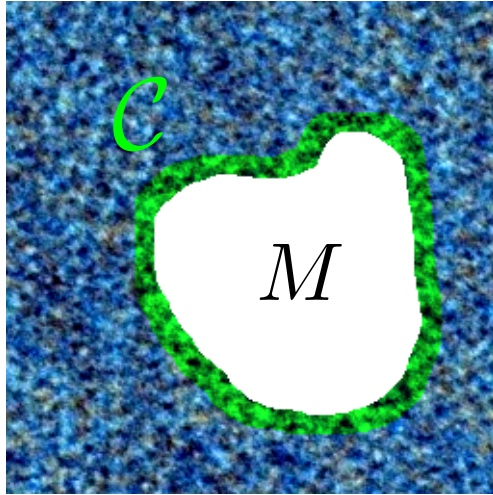


FIG. 1. *Textural inpainting via conditional simulation.* Inpainting with a stochastic texture model amounts to sampling the values on the mask M knowing the values on conditioning points C located at the border of the mask.

94 **1.2. Exemplar-based Inpainting, Sampling or Minimizing?...** An efficient
 95 way to model irregular images is to consider stochastic image models, and in partic-
 96 ular many texture synthesis algorithms can be formulated as sampling a probability
 97 distribution. Thus, one first strategy to inpaint textural parts of an image is to use
 98 an exemplar-based texture synthesis algorithm and to blend the synthesized content
 99 in the masked image. Such a method was proposed by Igehy and Pereira [40] who
 100 relied on Heeger-Bergen synthesis algorithm [38] to produce textural content.

101 On the other hand, if a stochastic image model is fixed, inpainting can be under-
 102 stood as sampling a conditional distribution, as illustrated on Fig. 1. This point of
 103 view was originally adopted by Efros and Leung [27]. These authors suggest approx-
 104 imate conditional sampling of a Markov random field (MRF) model by progressive
 105 completion of the unknown region using patch nearest neighbor search. Even if they
 106 show some texture inpainting results, their main concern is structured texture synthe-
 107 sis. For inpainting, this patch-based approach was precised in [11, 22]. In particular,
 108 Demanet et al. discuss the two possible formulations of the inpainting problem as
 109 either minimizing the energy E or sampling the probability distribution Ce^{-E} . They
 110 give several arguments to support that the variational point of view is a lighter and
 111 sufficient method to efficiently compute an inpainting solution. However, let us men-
 112 tion that the patch-based energy given in [22] is highly non-convex, and that the
 113 adopted optimization strategy does not offer much theoretical guarantees. Therefore,
 114 the empirical conclusions based on the results of this algorithm must be interpreted
 115 carefully. Our paper will shed some more light on this interesting (and still open)
 116 question, in the case of Gaussian textures.

117 Many other inpainting methods were inspired by these exemplar-based synthesis
 118 algorithms [22, 26, 21, 56, 43, 68, 4, 13, 70, 3, 2, 45, 50, 37, 55, 15]. These papers
 119 contain several clever algorithmic extensions of the original algorithm of [22]. In
 120 particular, Criminisi et al. [21] highlighted the importance of the pixel-filling order,
 121 and suggested that it should be driven by (progressively updated) patch priorities
 122 measuring the amount of available data and the quantity of structural information

123 in the currently synthesized content. Many authors [26, 43, 68, 3, 55] demonstrated
 124 that the inpainting problem could be more efficiently solved (both in visual terms or
 125 numerical terms) by relying on a multi-scale strategy. From a computational point
 126 of view, the speed of these algorithms highly depends on the method used for getting
 127 patch nearest neighbors, and many state of the art methods rely on the PatchMatch
 128 method which efficiently computes an approximate nearest neighbor field [6, 2, 50, 55].
 129 Let us also mention that the choice of the metric used for patch comparison may
 130 influence the inpainting results; to that purpose, the authors of [50, 55] suggested
 131 to improve the comparison by including textural features in the patch distance (e.g.
 132 local sum of absolute derivatives).

133 Here we would like to put the emphasis on a few papers which provide a thorough
 134 mathematical analysis of the variational formulation proposed by [22]. Aujol et al. [4]
 135 show the existence of a solution to a continuous analog of Demanet et al.’ energy
 136 among the set of piecewise roto-translations, propose several extensions of this prob-
 137 lem (allowing for either regularization or cartoon+texture decomposition), and also
 138 provide a 2D-example which illustrates the model ability to globally reconstruct ge-
 139 ometric features. Arias et al. [3] propose and compare several variational models
 140 obtained by varying the distance used in patch comparison (using the L^1 or L^2 norm
 141 on the image values or gradients), and also propose to replace the patch correspon-
 142 dence by generalized patch linear combinations using an adaptive weighting function.
 143 In [2], the same authors provide an additional mathematical analysis with a proof of
 144 the solution existence, of the convergence of the proposed minimization algorithm,
 145 and also a mathematical analysis of the PatchMatch algorithm. In these works, the
 146 inpainting problem is mainly formulated with a correspondence map (or a more gen-
 147 eral weighting function in [3]). In contrast, Liu and Caselles have shown in [50] that
 148 using an offset map instead allows to formulate inpainting as a discrete optimization
 149 problem which is efficiently solved with graph cuts. The statistics of patch offsets have
 150 been studied in [37]; He and Sun compute and exploit recurrent patch offsets in order
 151 to simplify the graphcut inpainting approach leading to an even faster algorithm.

152 Finally, the upper mentioned structural and exemplar-based methods can be com-
 153 bined to obtain hybrid structure-texture inpainting methods [9, 41, 63, 17]. Also, sev-
 154 eral authors proposed inpainting methods based on sparse decompositions of images
 155 or patches [28, 51, 16, 57]. In these methods, the inpainting is also formulated as a
 156 minimization problem (which can be coupled with the dictionary learning problem as
 157 in [51]). Although these methods are efficient in recovering missing data for thin or
 158 randomly-distributed masks, they are not able to fill large missing regions.

159 **1.3. Gaussian Conditional Simulation.** In this paper, we will address textu-
 160 ral inpainting by precise conditional sampling of a stochastic texture model.

161 In the computer graphics community, many authors have demonstrated the ex-
 162 pressive power of microtexture models based on Fourier phase randomization [47, 48]
 163 or on convolution of spot functions with noisy patterns [66]. Later, these models
 164 were studied in more detail by Galerne et al. [31] who propose in particular a simple
 165 analysis-synthesis pipeline for by-example microtexture synthesis with the Asymp-
 166 totic Discrete Spot Noise (ADSN) model (which is the Gaussian limit of Van Wijk’s
 167 Spot Noise model [66]). Such a Gaussian model is described by its first and second-
 168 order moments, and allows for fruitful mathematical developments, with applications
 169 in texture analysis [23], texture mixing [69], procedural texture synthesis [33, 32].

170 In this paper (following the preliminary work of [34]), we propose to take advan-
 171 tage of another benefit of the Gaussian model, which is the availability of a precise

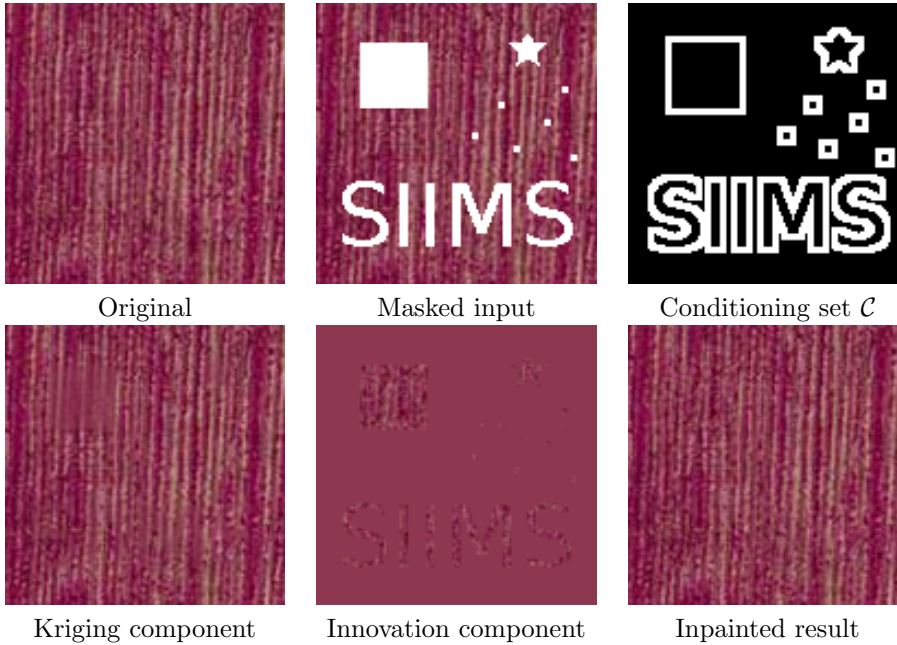


FIG. 2. **Summary of our microtexture inpainting method** The main idea of our method is to fill the masked region with a conditional sample of a Gaussian model. So this method is less about retrieving the initial image than computing another plausible sample of the texture model in the masked region. The Gaussian model is estimated from the unmasked values, and conditionally sampled knowing the values on a set 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.

172 conditional sampling algorithm. Indeed, for Gaussian vectors, independence is equiva-
 173 lent to uncorrelatedness, which can be rephrased as orthogonality in the Hilbert
 174 space of square-integrable random variables. Therefore, conditional simulation of a
 175 zero-mean Gaussian vector F only requires to compute an orthogonal projection F^*
 176 on a subspace of random variables (which corresponds to the conditional expectation
 177 given the known values) and to sample the orthogonal component $F - F^*$. Following
 178 the presentation of [44], we will rely on the terminology which is traditionally used
 179 in “simple kriging estimation”: the conditional expectation F^* will be called “kriging
 180 component”, and $F - F^*$ will be called “innovation component”. The role of these two
 181 components for conditional simulation is illustrated in Fig. 2. Let us mention that
 182 in the Gaussian case, solving the maximum *a posteriori* for the conditional model
 183 amounts to computing the conditional expectation (i.e. kriging component), which is
 184 very different from conditional sampling, as one can see on Fig. 2.

185 To the best of our knowledge, microtexture inpainting has not been addressed
 186 in those terms in the past. Gaussian conditional simulation algorithm was used by
 187 Hoffman and Ribak [39] for cosmological constrained simulations with parametric
 188 Gaussian models. More recently, local Gaussian conditional models were used for
 189 structured texture synthesis in [59, 58]. In the monoscale version [59], Raad et al.
 190 suggest to progressively sample the texture with conditional sampling of local Gaus-

191 sian models estimated from the exemplar (with nearest neighbor search as in [27, 67]);
 192 they also propose a multiscale adaptation of this algorithm [58]. As for [27], this algo-
 193 rithm could also be adapted for inpainting, but, because of the progressively estimated
 194 local models, the global model is not Gaussian.

195 Ordinary kriging was used by Chandra et al. [19] to interpolate sparsely sampled
 196 textural data. The first step of their procedure is to estimate a parametric Gaussian
 197 model by variogram fitting, and the second step is the computation of the kriging
 198 estimator (which is not a conditional sample). In comparison with our work, their
 199 estimation procedure can treat sparser data set (because the form of the variogram
 200 is imposed), but the interpolated image is not a satisfying texture sample because
 201 the innovation component is missing. Also, in the geostatistics literature, several
 202 authors have proposed generalized kriging algorithms for data prediction with various
 203 stochastic models [61, 20, 49]. In particular, in [60], Rue proposes a fast algorithm
 204 for conditional simulation in the particular case of Gaussian Markov random fields.
 205 Let us mention also that Almansa et al. [1] use generalized kriging for interpolation
 206 of digital elevation models (but do not address the parameter estimation issue). Also,
 207 let us refer the interested reader to [52] for a discussion on the links between texture
 208 synthesis and multiple-point geostatistics.

209 **1.4. Plan of the Paper.** In Section 2, we explain the traditional algorithm for
 210 Gaussian conditional simulation (using a terminology that is derived from kriging es-
 211 timation). In Section 3, we apply this conditional sampling algorithm to microtexture
 212 inpainting. In particular, we discuss the estimation of a Gaussian model on a masked
 213 exemplar, and we also provide a Fourier based algorithm which allows to compute
 214 the kriging estimation even when the number of conditioning points is very large.
 215 Finally, in Section 4, we provide several texture inpainting experiments to illustrate
 216 the validity of our approach; in particular we show that our method can compete with
 217 state of the art inpainting methods on textural content.

218 **2. Gaussian Conditional Simulation.** In this section, we recall the classical
 219 algorithm for conditional sampling of Gaussian random vectors. Following [44], we
 220 rely on a kriging framework that we introduce next.

221 **Notation.** Let Ω be a finite set. Let $(F(x))_{x \in \Omega}$ be a real-valued Gaussian
 222 vector, that is, a real-valued random vector for which any linear combination of the
 223 components is Gaussian. In this section, we assume that F has **zero mean**. The
 224 covariance of F is written

$$225 \quad (1) \quad \Gamma(x, y) = \text{Cov}(F(x), F(y)) = \mathbb{E}(F(x)F(y)), \quad x, y \in \Omega.$$

226 For a set $A \subset \Omega$ and a function $f : \Omega \rightarrow \mathbb{R}$ we denote by $|A|$ the cardinality of the
 227 finite set A , and $f|_A$ the restriction to A of the function f .

228 We also introduce a subset $\mathcal{C} \subset \Omega$ of conditioning points. Given prescribed values
 229 $\varphi : \mathcal{C} \rightarrow \mathbb{R}$ on \mathcal{C} , conditional Gaussian simulation consists in sampling the conditional
 230 distribution of F given that $F|_{\mathcal{C}} = \varphi$. As we shall see later, this conditional sampling
 231 makes sense as soon as φ belongs to the support of the distribution of $F|_{\mathcal{C}}$, which is
 232 the range of the restricted covariance matrix $\Gamma|_{\mathcal{C} \times \mathcal{C}}$ and denoted by $\text{Range}(\Gamma|_{\mathcal{C} \times \mathcal{C}})$.

233 **2.1. Simple Kriging Estimation.** We define the **simple kriging estimator**
 234 at $x \in \Omega$ as the conditional expectation

$$235 \quad (2) \quad F^*(x) = \mathbb{E}(F(x) \mid F(c), c \in \mathcal{C}).$$

236 This means that $F^*(x)$ is the best least-square estimation of $F(x)$ that can be obtained
 237 as a measurable function of $(F(c))_{c \in \mathcal{C}}$. A standard result of probability theory [24]

238 ensures that in the Gaussian case $F^*(x)$ is the orthogonal projection of $F(x)$ on the
 239 subspace of linear combinations of $(F(c))_{c \in \mathcal{C}}$ (for the L^2 -distance between square-
 240 integrable random variables). Hence, there exist coefficients $(\lambda_c(x))_{c \in \mathcal{C}}$ such that

$$241 \quad (3) \quad F^*(x) = \sum_{c \in \mathcal{C}} \lambda_c(x) F(c).$$

242 Such deterministic numbers $(\lambda_c(x))_{c \in \mathcal{C}}$ are called the **kriging coefficients**. Notice
 243 that by definition, $F^*(x) = F(x)$ for every $x \in \mathcal{C}$.

244 Generally speaking, for a given x , there may be several possible sets of kriging
 245 coefficients i.e. several vectors $(\lambda_c(x))_{c \in \mathcal{C}}$ which satisfy (3) (for example if there are
 246 two distinct points $c_1, c_2 \in \mathcal{C}$ such that $F(c_1) = F(c_2)$). But we will later give a
 247 canonical way to compute a valid set of kriging coefficients.

248 **2.2. Gaussian Conditional Sampling Using Kriging Estimation.** The fol-
 249 lowing theorem expresses Gaussian conditional sampling using the kriging estimator.
 250 Let us fix a set of coefficients $(\lambda_c(x))_{x \in \Omega, c \in \mathcal{C}}$ satisfying (3). For any $\varphi : \mathcal{C} \rightarrow \mathbb{R}$,
 251 we denote by φ^* the kriging estimation based on the values φ , defined for $x \in \Omega$
 252 by $\varphi^*(x) = \sum_{c \in \mathcal{C}} \lambda_c(x) \varphi(c)$. With a slight abuse of notation, if $\varphi : \Omega \rightarrow \mathbb{R}$, we will
 253 denote $\varphi^* = (\varphi|_{\mathcal{C}})^*$.

254 For the sake of completeness, we include the proof of the next standard result of
 255 probability theory (following [44]).

256 **THEOREM 1.** *F^* and $F - F^*$ are independent. Consequently, if G is independent*
 257 *of F and has the same distribution, then $H = F^* + (G - G^*)$ has the same distribution*
 258 *as F and satisfies $H|_{\mathcal{C}} = F|_{\mathcal{C}}$.*

259 *Proof.* Due to the orthogonal projection, $(F^*, F - F^*)$ is a Gaussian vector whose
 260 components F^* and $F - F^*$ are uncorrelated and thus independent. \square

261 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
 262 obtained with $\varphi^* + F - F^*$. In this decomposition, φ^* will be called the **kriging**
 263 **component** and $F - F^*$ will be called the **innovation component**.

264 **2.3. Expression of the Kriging Coefficients.** In order to compute the kriging
 265 estimator at a point $x \in \Omega$, one needs to compute the kriging coefficients $(\lambda_c(x))_{c \in \mathcal{C}}$
 266 introduced in (3). Recalling that $F^*(x)$ is the orthogonal projection of $F(x)$ on the
 267 subspace $\text{Span}(F(c), c \in \mathcal{C})$, we obtain

$$268 \quad (4) \quad \forall c \in \mathcal{C}, \quad \mathbb{E}(F^*(x)F(c)) = \mathbb{E}(F(x)F(c)) = \Gamma(x, c).$$

269 Substituting the expression (3) of $F^*(x)$ we get that $\lambda(x) = (\lambda_c(x))_{c \in \mathcal{C}}$ is a solution
 270 of the following $|\mathcal{C}| \times |\mathcal{C}|$ linear system

$$271 \quad (5) \quad \forall c \in \mathcal{C}, \quad \sum_{d \in \mathcal{C}} \lambda_d(x) \Gamma(d, c) = \Gamma(x, c),$$

272 which can be written in the more compact form $\lambda(x) \Gamma|_{\mathcal{C} \times \mathcal{C}} = \Gamma|_{\{x\} \times \mathcal{C}}$ using restric-
 273 tions of the covariance matrix Γ and where, by convention, we write the kriging
 274 coefficients $\lambda(x)$ as a row vector. Conversely, any solution of this linear system gives
 275 a valid set of kriging coefficients satisfying (3).

276 Aggregating the kriging coefficients in a $|\Omega| \times |\mathcal{C}|$ matrix $\Lambda = (\lambda_c(x))_{x \in \Omega, c \in \mathcal{C}}$, the
 277 system characterizing the kriging coefficients can also be written $\Lambda \Gamma|_{\mathcal{C} \times \mathcal{C}} = \Gamma|_{\Omega \times \mathcal{C}}$.
 278 If the matrix $\Gamma|_{\mathcal{C} \times \mathcal{C}}$ is invertible, the global system admits a unique solution $\Lambda =$

279 $\Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^{-1}$. In the case where $\Gamma_{|\mathcal{C} \times \mathcal{C}}$ is not invertible, it is always possible to compute
 280 valid kriging coefficients with the pseudo-inverse $\Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger$. Indeed, since the system (5)
 281 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
 282 the set of kriging coefficients given by

$$283 \quad (6) \quad \Lambda = \Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger.$$

284 Once a set Λ of valid kriging coefficients has been computed, a conditional sample
 285 of F given $F_{|\mathcal{C}} = \varphi$ can be obtained as

$$286 \quad (7) \quad \Lambda \varphi + F - \Lambda F_{|\mathcal{C}},$$

287 where φ and F are written as column vectors.

288 **Remark:** Let $x \in \Omega$ such that $F(x)$ is independent of $(F(c))_{c \in \mathcal{C}}$. Then the right-
 289 hand side of (5) vanishes so that 0 is a valid set of kriging coefficients for $F(x)$. This
 290 can also be seen on the conditional expectation: thanks to independence we have

$$291 \quad (8) \quad F^*(x) = \mathbb{E}(F(x) \mid F(c), c \in \mathcal{C}) = 0.$$

292 More precisely, 0 is trivially a minimal norm solution, which corresponds to our canon-
 293 ical choice $(\lambda_c(x))_{c \in \mathcal{C}} = 0 \times \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger$.

294 **2.4. Matrix Expression of the Conditional Simulation.** From this expres-
 295 sion of the conditional sample, we will derive the usual expression of the Gaussian
 296 conditional distribution in matrix notation (as e.g. in [61, 58]).

297 Let $p = |\mathcal{C}|$, $q = |\Omega \setminus \mathcal{C}|$ (where $\Omega \setminus \mathcal{C}$ denotes the complement of \mathcal{C} in Ω) and
 298 $n = |\Omega|$. Let us introduce the matrices $R = \begin{pmatrix} I_p & 0 \end{pmatrix} \in \mathbb{R}^{p \times n}$, $S = \begin{pmatrix} 0 & I_q \end{pmatrix} \in \mathbb{R}^{q \times n}$,
 299 Using the first p indices for the elements of \mathcal{C} , we can give block decompositions

$$300 \quad F = \begin{pmatrix} F_{|\mathcal{C}} \\ F_{|\Omega \setminus \mathcal{C}} \end{pmatrix} = \begin{pmatrix} RF \\ SF \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \Gamma_{|\mathcal{C} \times \mathcal{C}} & \Gamma_{|\mathcal{C} \times (\Omega \setminus \mathcal{C})} \\ \Gamma_{|(\Omega \setminus \mathcal{C}) \times \mathcal{C}} & \Gamma_{|(\Omega \setminus \mathcal{C}) \times (\Omega \setminus \mathcal{C})} \end{pmatrix} = \begin{pmatrix} R\Gamma R^T & R\Gamma S^T \\ S\Gamma R^T & S\Gamma S^T \end{pmatrix}.$$

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

$$303 \quad (9) \quad F \mid F_{|\mathcal{C}} = \varphi \sim \mathcal{N}(\Lambda \varphi, (I_n - \Lambda R) \Gamma (I_n - \Lambda R)^T).$$

304 In particular we get the usual formula

$$305 \quad (10) \quad \mathbb{E}(SF \mid F_{|\mathcal{C}} = \varphi) = S \Lambda \varphi = S \begin{pmatrix} R\Gamma R^T \\ S\Gamma R^T \end{pmatrix} (R\Gamma R^T)^\dagger \varphi = S\Gamma R^T (R\Gamma R^T)^\dagger \varphi.$$

306 For the conditional covariance, let us notice that the kriging system can be written
 307 as $\Lambda R \Gamma R^T = \Gamma R^T$ so that we have $(I - \Lambda R) \Gamma (I - \Lambda R)^T = \Gamma - \Lambda R \Gamma$, and thus the
 308 conditional covariance of SF is

$$309 \quad (11) \quad S(\Gamma - \Lambda R \Gamma) S^T = S\Gamma S^T - S\Lambda R \Gamma S^T = S\Gamma S^T - S\Gamma R^T (R\Gamma R^T)^\dagger R\Gamma S^T.$$

310 Notice that in the conditional distribution, RF is by definition considered as constant,
 311 and thus the conditional covariance of the whole vector F is

$$312 \quad (12) \quad \Gamma - \Lambda R \Gamma = \begin{pmatrix} 0 & 0 \\ 0 & S(\Gamma - \Lambda R \Gamma) S^T \end{pmatrix}.$$

313 When the matrix $R\Gamma R^T = \Gamma_{|\mathcal{C} \times \mathcal{C}}$ is non-singular, these expressions of the conditional
 314 expectation and variance are the same than in [61, 58].

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

315 **3. Microtexture Inpainting Algorithm.** This section contains our main con-
 316 tribution which is how to use Gaussian conditional sampling to perform microtexture
 317 inpainting.

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

326 After recalling the basics about the ADSN model, we discuss the estimation of
 327 such a model on a masked exemplar texture. Then we give an efficient and scalable way
 328 to compute the kriging estimator for the ADSN model by relying on conjugate gradient
 329 descent (numerical issues are discussed in the IPOL companion paper). Results of
 330 this algorithm on synthetic Gaussian textures and real images are given in the next
 331 section.

332 **3.1. ADSN Models.** As shown in [66, 31], a convenient model for microtexture
 333 is given by the asymptotic discrete spot noise (ADSN). Given a function $h : \mathbb{Z}^2 \rightarrow \mathbb{R}$
 334 with finite support, the ADSN associated to h is the convolution of h with a normalized
 335 Gaussian white noise W on \mathbb{Z}^2 , defined as

$$336 \quad (13) \quad \forall x \in \mathbb{Z}^2, \quad h * W(x) = \sum_{y \in \mathbb{Z}^2} h(y)W(x - y).$$

337 This Gaussian random field is stationary, has zero mean, and its covariance function
 338 is given by

$$339 \quad (14) \quad \forall x, y \in \mathbb{Z}^2, \quad \mathbb{E}(h * W(x)h * W(y)) = \sum_{z \in \mathbb{Z}^2} h(x - z)h(y - z) = (h * \tilde{h})(x - y),$$

340 where $\tilde{h}(z) = h(-z)$. The restriction on a finite $\Omega \subset \mathbb{Z}^2$ of $h * W$ is a zero-mean Gaus-
 341 sian model $(F(x))_{x \in \Omega}$. Thanks to the simple convolutive expression of the ADSN,
 342 it can be efficiently sampled using the fast Fourier transform (FFT). Indeed, if we
 343 assume that the support of h is included in Ω , we can extend h by zero-padding to
 344 a twice larger domain $\bar{\Omega}$, convolve with a Gaussian white noise on $\bar{\Omega}$ with periodic
 345 boundary conditions, and then crop the result on Ω . Similarly, the covariance func-
 346 tion Γ of U can be efficiently computed with a periodic convolution of zero-padding
 347 extensions of h and \tilde{h} .

348 Alternatively, one can also consider a periodic ADSN model on the domain Ω .
 349 Given a function $h : \Omega \rightarrow \mathbb{R}$, the periodic ADSN model is the convolution of h with
 350 a normalized Gaussian white noise W on Ω with periodic boundary conditions. The
 351 simulation of the periodic model is even easier since it only involves one periodic
 352 convolution (without need of prior zero-padding extension or posterior restriction).
 353 Apart from this slight gain of complexity, there is no general reason to favor the
 354 periodic model. The choice is often driven by the context of application; for example,
 355 non-periodic models are better suited for on-demand texture synthesis [33, 32]. We
 356 refer the interested reader to Chapter 2 of [46] for a fully detailed exposure regarding
 357 both ADSN models.

358 **Extension to Color Images.** ADSN models extend to color images by con-
 359 volving each color channel with the same white noise in (13). This gives an \mathbb{R}^d -valued
 360 Gaussian random field F on Ω (where d is the number of channels, i.e. 3 for color
 361 images). Regarding the conditional simulation, a simple way to understand this ex-
 362 tension is to consider the \mathbb{R}^d -valued random field F as a real-valued random field on
 363 $\Omega \times \{1, \dots, d\}$. The covariance matrix is then given by

$$364 \quad (15) \quad \forall (x, j), (y, k) \in \Omega \times \{1, \dots, d\}, \quad \Gamma((x, j), (y, k)) = \mathbb{E}(F_j(x)F_k(y)).$$

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

368 **3.2. Estimation of the Gaussian Model.** If the image $u : \Omega \rightarrow \mathbb{R}^d$ were
 369 entirely available, the estimation procedure would be the same as for texture syn-
 370 thesis [31, 33], which is briefly recalled here. The expectation is estimated by $\bar{u} =$
 371 $\frac{1}{|\Omega|} \sum_{x \in \Omega} u(x)$. Then, from $u - \bar{u}$ we derive an ADSN model by computing the nor-
 372 malized spot $t_u = \frac{1}{\sqrt{|\Omega|}}(u - \bar{u})$ (extended by zero-padding). The microtexture u is
 373 then synthesized by sampling $\bar{u} + t_u * W$, with W a normalized Gaussian white noise.

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

$$377 \quad (16) \quad \bar{v} = \frac{1}{|\omega|} \sum_{x \in \omega} v(x), \quad t_v(x) = \begin{cases} \frac{1}{\sqrt{|\omega|}}(v(x) - \bar{v}) & \text{if } x \in \omega, \\ 0 & \text{otherwise.} \end{cases}$$

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

$$379 \quad (17) \quad c_v(h) = \frac{1}{|\omega|} \sum_{x \in \omega \cap (\omega - h)} (u(x+h) - \bar{v})(u(x) - \bar{v})^T \in \mathbb{R}^{d \times d}.$$

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

386 We would like to point out here that designing more precise estimators of the
 387 covariance is an interesting question. In particular, at first sight one can be puzzled
 388 by the normalization of (17). A better normalized estimator $c'_v(h)$ would be obtained
 389 by replacing $\frac{1}{|\omega|}$ by $\frac{1}{|\omega \cap (\omega - h)|}$ in this formula. But a drawback of this new estimator is
 390 that it does not define a semi-definite positive estimator, and thus is not associated to
 391 a Gaussian model that could be sampled. A way to cope with this effect is to enforce
 392 semi-definite positiveness, which in the stationary case is equivalent to project on
 393 the non-negative orthant in Fourier domain. We have led some experiments in this
 394 direction, and they have shown that the resulting Gaussian model is not better than
 395 the one obtained with the naive estimator (both in terms of resynthesis or in terms
 396 of optimal transport distance between Gaussian models [33]). Indeed, the projection
 397 on the Fourier orthant has a dramatic impact on the model (in particular, it may
 398 significantly impact the estimation of the marginal variance).

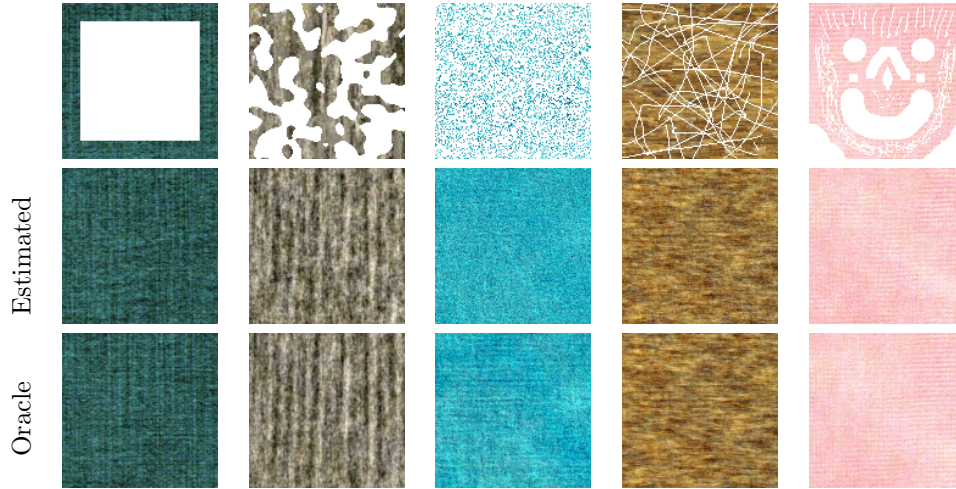


FIG. 3. *Estimation of an ADSN model on a masked exemplar. We illustrate with several types of mask the estimation of the Gaussian model with the naive estimator (17) using $\omega = \Omega \setminus M$. We display in the first row the masked exemplar, in the second row a sample of the estimated ADSN 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).*

399 One explanation of the success of the naive estimator for regular masks is that
 400 in this case we have $\frac{|\omega \cap (\omega - h)|}{|\omega|} \approx 1$ when $h \approx 0$. Therefore the naive estimator is
 401 approximately well normalized around 0 and thus correctly estimates the covariance
 402 in a neighborhood of 0, which is the most important part for microtexture images.
 403 It might be possible to design more faithful estimators (e.g. using an expectation-
 404 maximization algorithm), but this challenging problem is beyond the scope of this
 405 paper.

406 **3.3. Kriging Estimation with Conjugate Gradient Descent.** In this sec-
 407 tion, we propose an efficient way to compute a conditional sample of the ADSN model.
 408 The most difficult part consists in solving a large linear system involving the condi-
 409 tional values. This step is dealt with by using a conjugate gradient descent algorithm,
 410 which proves to be efficient even for very large images.

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

417 As explained in the last section, after subtracting the estimated mean \bar{v} , we
 418 can use the ADSN model $(F(x))_{x \in \Omega}$ associated to the spot t_v (which is a zero mean
 419 Gaussian vector). Using the framework of Section 2, we are able to draw a conditional
 420 sample $(F(x))_{x \in \Omega}$ given $F|_{\mathcal{C}} = u|_{\mathcal{C}} - \bar{v}$. With the notation of Section 2, such a
 421 conditional sample is given by

$$422 \quad (18) \quad (u - \bar{v})^* + F - F^* = \Lambda((u - \bar{v})|_{\mathcal{C}}) + F - \Lambda(F|_{\mathcal{C}}).$$

423 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 $\varphi \in \mathbb{R}^{\mathcal{C}}$.

424 Let us begin with the multiplication by $\Gamma_{|\Omega \times \mathcal{C}}$, which is easier. Assume that
 425 $\psi = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger \varphi$ has been computed. Using the notation of Section 2.4, $\Gamma_{|\Omega \times \mathcal{C}} \psi = \Gamma \Psi$,
 426 where $\Psi = R^T \psi \in \mathbb{R}^\Omega$ is the zero-padding extension of ψ . Now, since Γ is the
 427 covariance function of an ADSN model, it can be simply computed by convolution.
 428 More precisely, $\Gamma \Psi$ is the restriction on Ω of the convolution of Ψ by $t_v * \tilde{t}_v$ (with zero-
 429 boundary condition in the non-periodic ADSN, and periodic boundary conditions in
 430 the periodic ADSN).

431 Computing $A^\dagger \varphi$ where $A = \Gamma_{|\mathcal{C} \times \mathcal{C}}$ is more costly. Assume for a moment that A
 432 is invertible. Then computing $A^{-1} \varphi$ amounts to solving a linear system of size $p \times p$
 433 (where $p = d|\mathcal{C}|$). Since A is symmetric positive-definite, this can be reduced to solv-
 434 ing two triangular systems thanks to the Cholesky factorization of A . Nevertheless,
 435 finding the Cholesky factorization of A requires $\mathcal{O}(p^3)$ flops in general. Therefore,
 436 this direct method will only work for small values of p . This was a major limitation
 437 of our preliminary work presented in [34].

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

444 Following [42], we compute $A^\dagger \varphi$ by performing a conjugate gradient descent on

$$445 \quad (19) \quad f : \psi \mapsto \frac{1}{2} \|A\psi - \varphi\|^2$$

446 with initialization $\psi_0 = 0$. This optimization procedure actually solves the normal
 447 equations $A^T A \psi = A^T \varphi$, which are equivalent to $A\psi = \varphi$ when $\varphi \in \text{Range}(A)$ (recall
 448 that the range of A and the kernel of A^T are orthogonal subspaces). The algorithm
 449 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
-

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

454 Since the multiplication by A can be computed efficiently with the FFT, the
 455 complexity of Algorithm CGD with N iterations is $\mathcal{O}(N|\Omega| \log |\Omega|)$. The main benefit
 456 of using this algorithm is that it allows to consider very large conditioning sets \mathcal{C} .
 457 Of course, increasing \mathcal{C} may increase the number of required iterations to obtain the
 458 solution at a given precision ε . But if the condition number of the system is low, we

459 will get a good approximation of the solution in a reasonable number of iterations. Let
 460 us mention that Algorithm CGD is theoretically expected to get the exact solution
 461 in a finite number of iterations, but this remark is not useful for our practical case
 462 because of the numerical errors caused by the FFT.

463 **Stopping criterion.** The stopping criterion that we use in Algorithm CGD is
 464 $\|r_k\| \leq \varepsilon$ where the residual at iteration k is given by

$$465 \quad (20) \quad r_k = A^T \varphi - A^T A \psi_k,$$

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

470 3.4. Comments on the Kriging System.

471 **The matrix A is not necessarily invertible.** Indeed, let us consider the case
 472 of a color periodic ADSN model on Ω estimated by (16). Then the DFT of the
 473 covariance operator Γ is given by

$$474 \quad (21) \quad \widehat{t}_v(\xi) \widehat{t}_v(\xi)^* = \begin{cases} \frac{1}{|\omega|} \widehat{v}(\xi) \widehat{v}(\xi)^* & \text{if } \xi \neq 0 \\ 0 & \text{if } \xi = 0 \end{cases}.$$

475 As noted in [69], this matrix has rank ≤ 1 which constrains the rank of the ma-
 476 trix Γ (of size $d|\Omega| \times d|\Omega|$) to be bounded by $|\Omega| - 1$. Since A is a submatrix of Γ ,
 477 $\text{Rank}(A) \leq |\Omega| - 1$. In particular, if the conditioning set is sufficiently big so that
 478 $d|\mathcal{C}| \geq |\Omega|$, then A cannot be invertible.

479 **The vector $\varphi = u_{|\mathcal{C}} - \bar{u}$ may not be in the range of A .** Indeed, if A is not
 480 invertible, the conditioning values could be out of the range of A . However, if the
 481 masked texture has been drawn from the oracle Gaussian distribution $\mathcal{N}(0, \Gamma)$ (as in
 482 the experiments of Section 4.1), then $\varphi \sim \mathcal{N}(0, \Gamma_{|\mathcal{C} \times \mathcal{C}})$ is almost surely in the range
 483 of $A = \Gamma_{|\mathcal{C} \times \mathcal{C}}$. Anyway, $\varphi \notin \text{Range}(A)$ is not a problem for applying Algorithm CGD
 484 because taking $A\varphi$ implicitly cancels the component on the kernel of A .

485 Notice also that if the estimated ADSN model is well adapted to the masked
 486 texture, then it is likely that φ is close to the range of A . In practice, the distance
 487 of φ to the range of A is bounded by the norm of the residual obtained with the direct
 488 conjugate gradient method:

$$489 \quad (22) \quad \|\varphi - A\psi_k\| \geq \text{dist}(\varphi, \text{Range}(A)).$$

490 **3.5. Complete Algorithm.** To end this section, we summarize our microtex-
 491 ture inpainting algorithm. In Algorithm CGD the matrix $A = \Gamma_{|\mathcal{C} \times \mathcal{C}}$ is not formed
 492 explicitly, and we only need to apply it efficiently with the FFT-based algorithm.
 493 Also, if one is not interested in the kriging and innovation components but only in
 494 the inpainting result, then only one instance of gradient descent is needed since the
 495 output only depends on

$$496 \quad (23) \quad (u - \bar{v} - F)^* = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger (u_{|\mathcal{C}} - \bar{v} - F_{|\mathcal{C}}).$$

497 The overall complexity of this algorithm is $\mathcal{O}(k_{\max} |\Omega| \log |\Omega|)$ where k_{\max} is the
 498 number of iterations used in the gradient descent algorithm. The overall number of

499 FFTs required by the whole inpainting process (whose detailed computation can be
 500 found in the IPOL companion paper) is $(4k_{\max} + 6)d$ FFTs. Using our C implemen-
 501 tation (involving parallel computing, in particular for the FFT) run with a modern
 502 computer (Intel i7 processor @2.60GHz with 4 cores), the whole inpainting process
 503 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 $(u - \bar{v})^* = t_v * \tilde{t}_v^T * \Psi_1$ (kriging component)
 $F^* = t_v * \tilde{t}_v^T * \Psi_2$ (innovation component)

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

504 **4. Results and Discussion.**

505 **4.1. Inpainting with an Oracle Model.** First, we propose a validation exper-
 506 iment to confirm that Gaussian conditional simulation can be applied to constrained
 507 microtexture synthesis. For that, we consider a non-masked texture image u on which
 508 we estimate an oracle ADSN model as explained in Section 3.2. We compute one real-
 509 ization of this oracle ADSN model (with a random seed s_1), on which we put a mask
 510 M . Then we perform conditional sampling of the values in the masked region (with
 511 a random seed $s_2 \neq s_1$), based on a set of conditioning points \mathcal{C} , which is taken to be
 512 either $\mathcal{C} = \Omega \setminus M$ or $\mathcal{C} = \partial_3 M$. This amounts to applying our inpainting algorithm,
 513 except that we use an oracle model.

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

526 Let us remark that the results obtained in Fig. 5 with irregular masks look im-

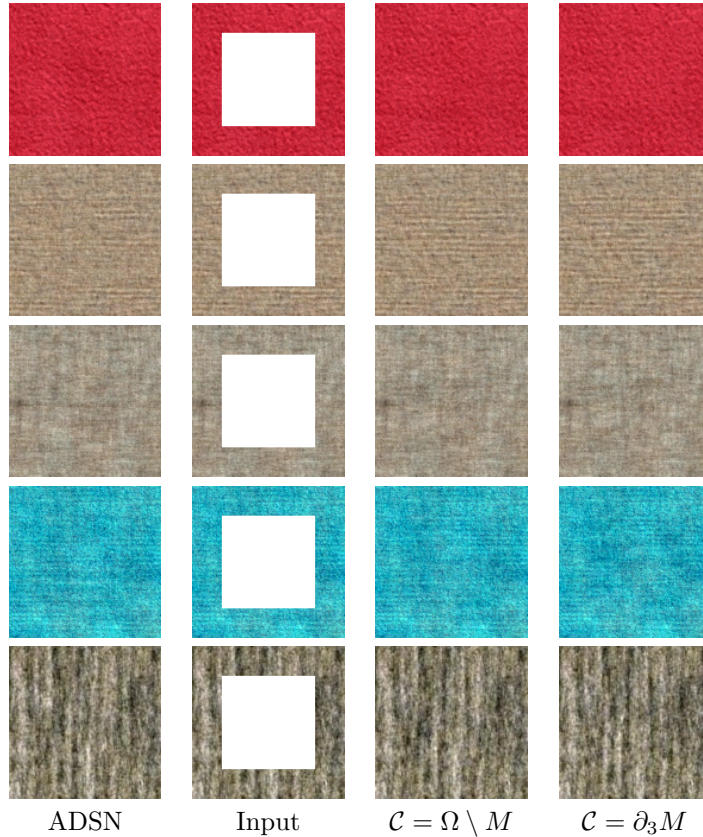


FIG. 4. *Inpainting Gaussian textures with the oracle Gaussian model - regular masks.* The masked input has been inpainted with Gaussian conditional simulation using an oracle Gaussian model (estimated from the unmasked exemplar texture) based on conditioning values on $C \subset \Omega$. From left to right, we show a sample of the oracle model, the masked input, and the inpainted results obtained for $C = \Omega \setminus M$ or $C = \partial_3 M$. 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 $C = \Omega \setminus M$ and $C = \partial_3 M$ which shows that conditioning on $C = \partial_3 M$ is practically sufficient.

527 pressive at first sight since a wide majority of pixels are masked; but one should recall
 528 that in this experiment the oracle ADSN model is estimated on the unmasked exemplar,
 529 which makes the inpainting problem much simpler (compare with the results of
 530 Section 4.2).

531 In the experiment of Fig. 6, we show that Gaussian conditional simulation with an
 532 oracle model can be used to extrapolate textural content defined on a thin domain. In
 533 this case, the simulated conditional Gaussian vector is very high-dimensional, which
 534 illustrates the benefit of having a scalable algorithm based on gradient descent (and
 535 not on explicit computation of the covariance operators).

536 **4.2. Inpainting with an Estimated Gaussian Model.** In this section, we
 537 provide experimental results which show that our algorithm is able to inpaint holes
 538 in microtextures, whatever the size of the hole, and with only minimal requirements
 539 on the hole regularity. In contrast with the last section, the Gaussian model is now
 540 estimated from the masked exemplar. We will show that the naive estimation tech-
 541 nique explained in Section 3.2 and illustrated in Fig. 3 leads to satisfying inpainting



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).

542 results except in the case where the mask is made of randomly scattered pixels. In
 543 all the experiments shown in this section, we took $\mathcal{C} = \partial_3 M$.

544 Let us first comment the results of Fig. 7, which were partially available in the
 545 preliminary version of this work [34]. In this preliminary work, we suggested to
 546 manually choose a rectangle subdomain ω for the estimation of the Gaussian model.
 547 In the following figures, the manually chosen subdomains ω are displayed with a red
 548 box on the masked texture. Now we also apply the automatic naive procedure which
 549 consists in taking $\omega = \Omega \setminus M$. As one can see in Fig. 7, these first inpainting
 550 results are satisfying (the holes can hardly be distinguished in the inpainted image). Also,
 551 one can observe that the results are comparable when using the manually chosen ω
 552 or the canonical choice $\omega = \Omega \setminus M$. So when inpainting an image which is made of
 553 only one texture, setting $\omega = \Omega \setminus M$ is always a reasonable choice.

554 In Fig. 8, we show some results of our algorithm for several microtextures and
 555 macrotextures, with various types of masks. As one can observe, the results with
 556 microtextures are globally very satisfying; the most difficult case being the irregular
 557 mask of the third column, for which the Gaussian model cannot be properly estimated.
 558 Surprisingly, we also obtained quite convincing results on more structured textures.

559 To end this section, we show that our algorithm can be used to inpaint textural
 560 parts of more general images. For example, on Fig. 9, we used it to remove some
 561 undesirable details located in a region composed of one homogeneous microtexture. In
 562 such a case, one must manually specify the subdomain ω on which the Gaussian model
 563 is estimated in order to take only values in the desired texture region. Of course, the
 564 mask must also be provided by the user. But after that, our algorithm can be applied,
 565 without needing any additional parameter. One nice feature of our algorithm is that
 566 the synthesized texture is naturally blended in the surrounding content, which is not
 567 necessarily the case with other state of the art inpainting techniques (see Section 4.5
 568 for a more detailed comparison).

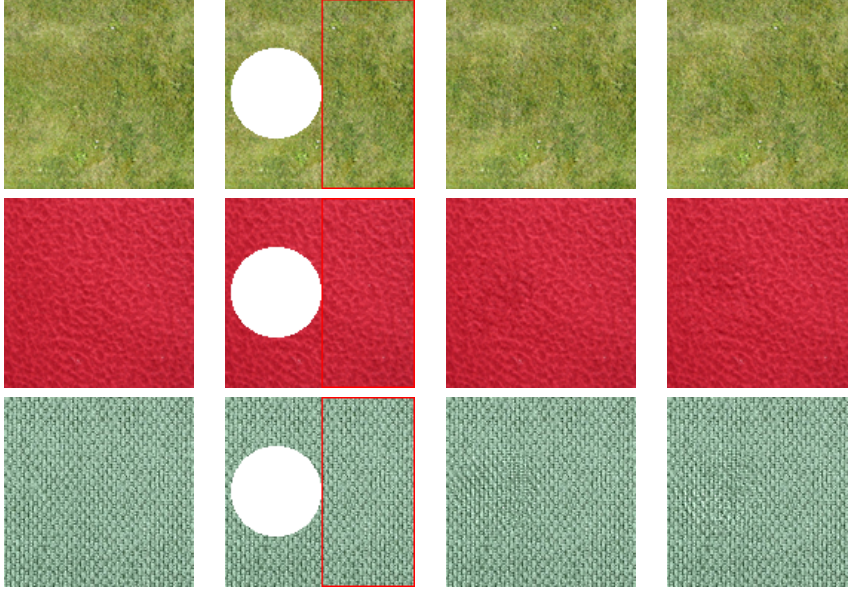


FIG. 7. *Examples of microtexture inpainting - circular holes. From left to right, we display an original microtexture, a masked version with a circular hole, the inpainting result obtained with a manually chosen ω , and the inpainted result with the automatic choice $\omega = \Omega \setminus M$. The manually chosen subdomain ω (which serves for the estimation of the ADSN model) is delimited by a red box on the masked exemplar. The inpainting is satisfactory because the output has similar aspect than the original unmasked texture while being different on the mask. Notice that both manual and automatic choices of ω give very similar results.*

569 **4.3. Computing and Visualizing the Kriging Coefficients.** In order to
 570 better understand the conditional simulation, it is interesting to visualize the kriging
 571 coefficients. Heuristically speaking, every non-zero coefficient $\lambda_c(x)$ corresponds to
 572 a position x whose value $F(x)$ depends on $F(c)$ in the conditional simulation. We
 573 can thus expect the correlations of the adopted Gaussian model to be reflected in the
 574 kriging coefficients.

575 First, let us explain how to visualize $(\lambda_c(x))_{x \in \Omega}$ for a fixed $c \in \mathcal{C}$. Since $\Lambda =$
 576 $\Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger$, we have

$$577 \quad (24) \quad (\lambda_c(x))_{x \in \Omega} = \Lambda \delta_c = \Gamma_{|\Omega \times \mathcal{C}} \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger \delta_c,$$

578 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
 579 use our algorithm on a Dirac input.

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

$$582 \quad (25) \quad (\lambda_c(x))_{c \in \mathcal{C}} = \Lambda^T \delta_x = \Gamma_{|\mathcal{C} \times \mathcal{C}}^\dagger \Gamma_{|\mathcal{C} \times \Omega} \delta_x,$$

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

586 In the case of the inpainting application, we get the coefficients shown in Fig. 10.
 587 These results clearly indicate that the correlations captured in the Gaussian model
 588 are reflected by the large kriging coefficients. We can also observe on this figure that
 589 the kriging coefficients are not positive in general.

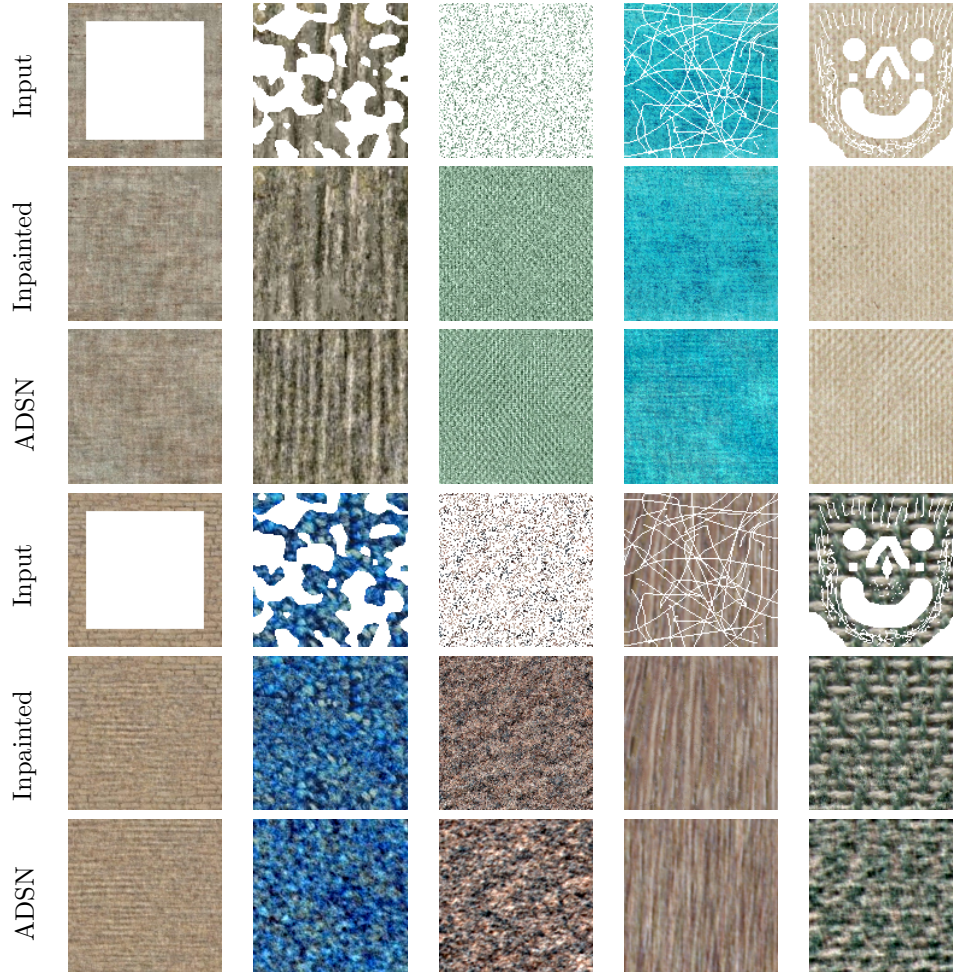


FIG. 8. *Examples of textural inpainting.* We present results of our inpainting method for several textures and masks. From top to bottom (rows 1-3 and rows 4-6), we display a masked input, the inpainted result, and a sample of the estimated ADSN model (which is useful to exhibit the limit of the Gaussian 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.

590 **4.4. Impact of the Size of the Conditioning Border.** In this section, we
 591 investigate the impact of changing the size of the conditioning border. Again, an ideal
 592 setting would be to choose $\mathcal{C} = \Omega \setminus M$, but then the kriging system is very large. Here
 593 we will confirm that taking $\mathcal{C} = \partial_w M$ is sufficient, and we will precisely examine the
 594 variation of the conditional model when increasing the width w of the border.

595 In order to give a quantitative comparison, we suggest to compute distances
 596 between the conditional models, which are basically Gaussian random vectors on M .
 597 A possible way to perform this comparison is to rely on the L^2 -optimal transport
 598 distance between Gaussian models; this distance has already been used in several

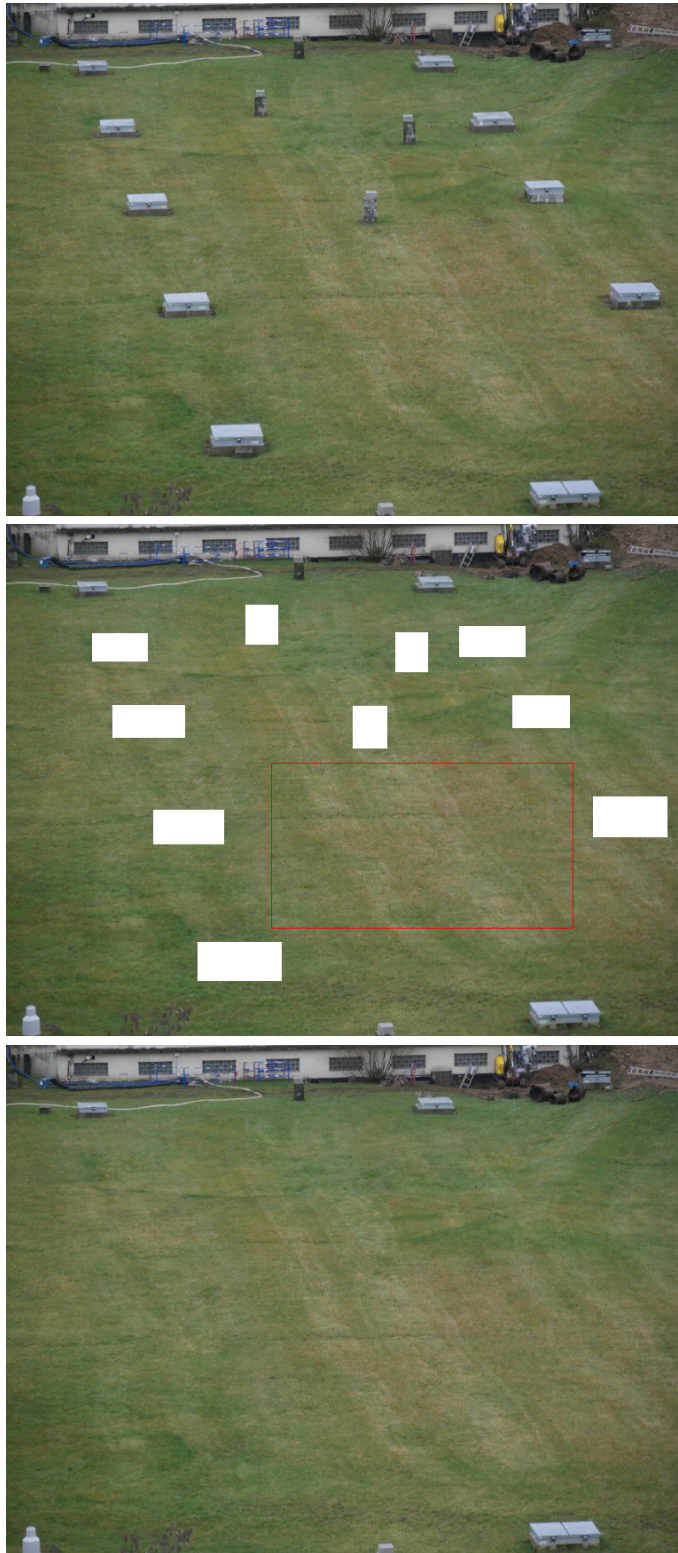


FIG. 9. *Inpainting textural parts of an image.* From top to bottom, we display the original image (of size 768×577), the masked input (the Gaussian model has been estimated in the subdomain ω delimited by the red box), and the inpainted result. Our algorithm is able to synthesize microtexture content which naturally blends with the surrounding context.

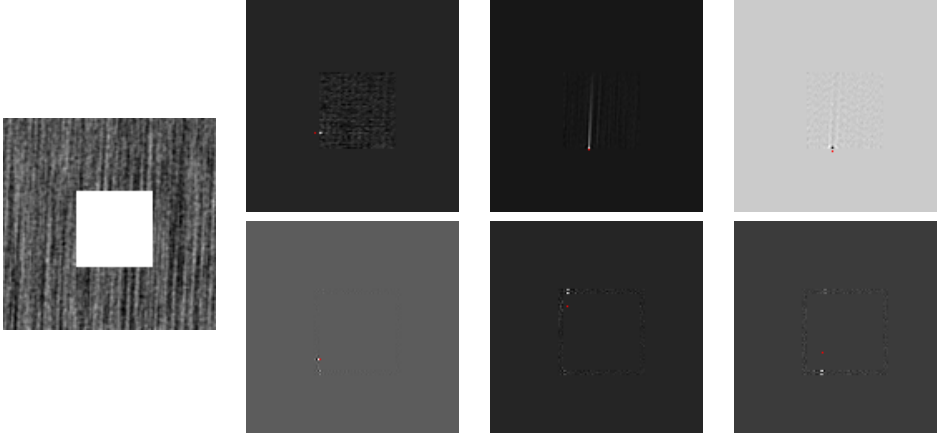


FIG. 10. *Visualizing Kriging coefficients.* In the first column, we display the masked input. For the three other columns: in the first row, we display the kriging coefficients $(\lambda_c(x))_{x \in M}$ for different positions of the conditioning pixel $c \in \mathcal{C}$ (drawn in red); in the second 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.

599 works about texture synthesis [69, 33]. Let us recall that, given two Gaussian models
 600 $\mu_X = \mathcal{N}(m_X, \Sigma_X), \mu_Y = \mathcal{N}(m_Y, \Sigma_Y)$, this distance is defined as

$$601 \quad (26) \quad d_{\text{OT}}(\mu_X, \mu_Y) = \inf \mathbb{E}(\|F - G\|^2)$$

602 where $\|\cdot\|$ is the standard Euclidean norm, and where the infimum extends over all
 603 couples of random vectors (F, G) such that $F \sim \mathcal{N}(m_X, \Sigma_X)$ and $G \sim \mathcal{N}(m_Y, \Sigma_Y)$.
 604 As shown in [25], in the Gaussian case, this distance is given by the explicit formula

$$605 \quad (27) \quad d_{\text{OT}}(\mu_X, \mu_Y)^2 = \|m_X - m_Y\|^2 + \text{Tr}(\Sigma_X) + \text{Tr}(\Sigma_Y) - 2\text{Tr}\left((\Sigma_X \Sigma_Y)^{1/2}\right).$$

606 We will use this distance to compare the conditional models obtained with sev-
 607 eral conditioning sets \mathcal{C} . More precisely, we consider a gray-level exemplar texture
 608 $u : \Omega \rightarrow \mathbb{R}$ on which we estimate an oracle model $\mathcal{N}(\bar{u}, \Gamma)$ and on which we put a
 609 mask $M \subset \Omega$. Then, we consider the reference conditional model $\mu_\infty = \mathcal{N}(m_\infty, \Sigma_\infty)$
 610 obtained with $\mathcal{C}_\infty = \Omega \setminus M$, and the conditional models $\mu_w = \mathcal{N}(m_w, \Sigma_w)$ obtained
 611 with $\mathcal{C}_w = \partial_w M$ (border of M with width w pixels). Using the expressions found in
 612 Section 2.3 and Section 2.4, we recall

$$613 \quad m_w = \Gamma_{|M \times \mathcal{C}_w} \Gamma_{|\mathcal{C}_w \times \mathcal{C}_w}^\dagger (u - \bar{u})_{|\mathcal{C}_w}, \quad \Sigma_w = \Gamma_{|M \times M} - \Gamma_{|M \times \mathcal{C}_w} \Gamma_{|\mathcal{C}_w \times \mathcal{C}_w}^\dagger \Gamma_{|\mathcal{C}_w \times M}.$$

614 For our experiment, we choose a reasonably small texture so that all these covariance
 615 matrices can be explicitly built and stored (relying on standard numerical routines for
 616 pseudo-inverse and square roots computation²). Once computed the Gaussian models

²The pseudo-inverse is only computed up to a given precision. But, following the remark at

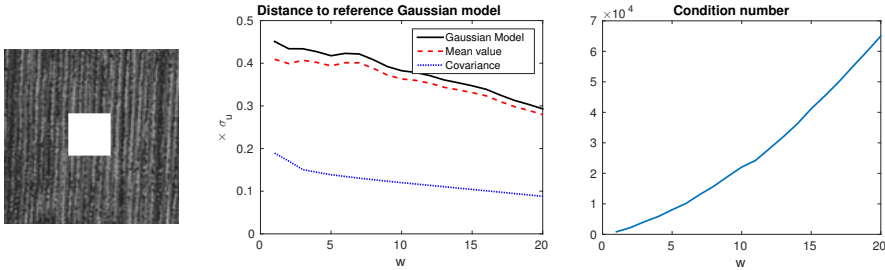


FIG. 11. *Quantitative study of the conditional models depending on the conditioning set \mathcal{C} . We computed the distance between the reference conditional model (obtained for $\mathcal{C}_\infty = \Omega \setminus M$) and the conditional models (obtained for $\mathcal{C}_w = \partial_w M$), see (28). 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.*

617 $\mu_w = \mathcal{N}(m_w, \Sigma_w)$, we plot the function

618 (28)
$$w \in \{1, \dots, 20\} \mapsto \frac{d_{\text{OT}}(\mu_w, \mu_\infty)}{\sigma_u \sqrt{|M|}},$$

619 where σ_u is the marginal standard deviation of the oracle model. Notice that we
 620 normalize the distance by $\sigma_u \sqrt{|M|}$ in order to get a number which reads as a single-
 621 pixel value on a scale that is adapted to the expected variation of the texture model.
 622 We also report separately the distances between the mean values and the covariance
 623 matrices, i.e.

624
$$d(m_w, m_\infty) = \|m_w - m_\infty\|,$$

 625
$$d(\Sigma_w, \Sigma_\infty)^2 = \text{Tr}(\Sigma_w) + \text{Tr}(\Sigma_\infty) - 2\text{Tr}\left((\Sigma_w \Sigma_\infty)^{1/2}\right),$$

 626

627 The results can be observed in Fig 11. One can observe a global tendency of these
 628 distances to decrease when the conditioning border gets larger. But we do not observe
 629 a sudden plunge of the value (even if the covariance distance decreases a bit quicker
 630 for $w < 5$). Also, an interesting fact raised by these graphs is that the marginal
 631 error made when replacing \mathcal{C}_∞ by \mathcal{C}_w is in general less than one σ_u . Notice also that
 632 when w increases, the kriging system become more and more ill-conditioned, which
 633 corroborates the numerical results given in Section 3.4.

634 We also propose in Fig. 12 a more qualitative experiment. This qualitative study
 635 is important to examine the quality of the inpainting result around the mask border
 636 (which is not reflected through the marginal L^2 error between two conditional models).
 637 For several values of the border width $w = 1, 3, 5$, we inpaint a texture image (with
 638 the oracle Gaussian model), and we compare the results with the one obtained in the
 639 ideal case $\mathcal{C}_\infty = \Omega \setminus M$. In order to give per-pixel comparison, we used the same
 640 random seed for the conditional sampling. Apart from the visual results, we also
 641 report the distance between the mean values of the corresponding conditional models
 642 (normalized by $\sigma_u \sqrt{|M|d}$ where d is the number of channels).

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} .

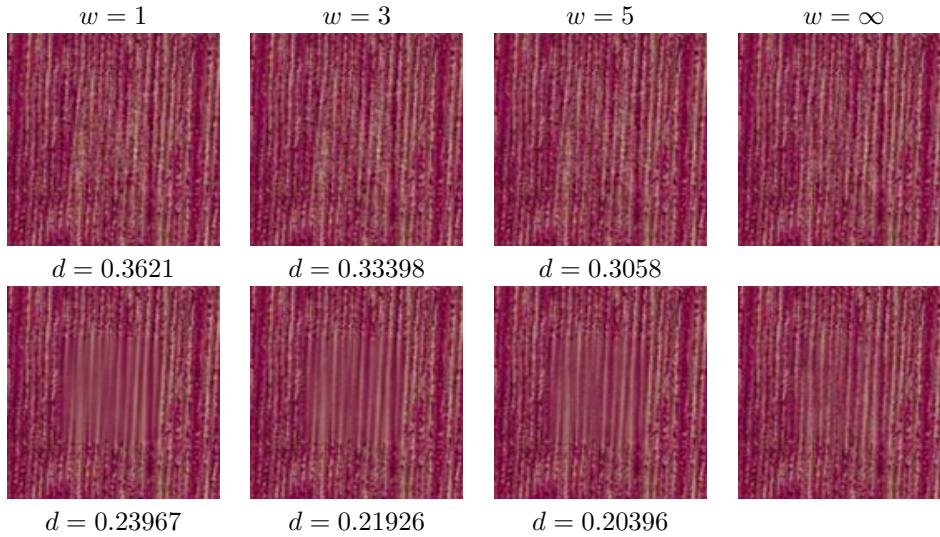


FIG. 12. *Qualitative study of the conditional models depending on the conditioning set \mathcal{C} . From left to right, we display the inpainting results obtained for \mathcal{C} being a border of M of width $w = 1, 3, 5$ pixels, and also the limit solution $\mathcal{C} = \Omega \setminus M$. In the 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.*

643 It is interesting to notice that the kriging components look very different with
 644 $w = 5$ and $w = \infty$. Indeed, when the conditioning set gets larger, the kriging
 645 component depends on a larger number of random variables, and thus has an increased
 646 stochastic nature. This explains why the distance between the Gaussian models (or
 647 their mean or covariance functions) does not quickly tend to zero when w increases.
 648 Still, as reflected by the example of Fig. 12 and as observed in all our experiments,
 649 the inpainting result is already good for $w = 3$ (in particular, for many textures, this
 650 value is sufficient to naturally blend the inpainted domain in the context).

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

654 **4.5. Comparisons.** In this section, we compare our microtexture inpainting
 655 algorithm with many recent inpainting techniques, and in particular with patch-based
 656 state-of-the-art methods.

657 First, in Fig. 13, we compare our method with two very famous methods, namely,
 658 total variation (TV) based inpainting [18], and the patch-based method of Criminisi
 659 et al [21]. As could be expected, the TV inpainting method is not appropriate for
 660 this example, because the water texture in this image is not of bounded variation.
 661 In contrast, much better results are obtained with our method or the one of Crimi-
 662 nisi et al. Compared to [21], our result seems a bit more stochastic, maybe even
 663 too stochastic in the upper part of the inpainted domain. This clearly reflects one
 664 limitation of our model, which is stationarity. Indeed, the water texture outside the
 665 mask is not exactly stationary since it exhibits less variation in the top of the image.

666 On Fig. 14 and Fig. 15, we compare our Gaussian inpainting algorithm with
 667 several patch-based methods. On the first rows of Fig. 14, one can observe that

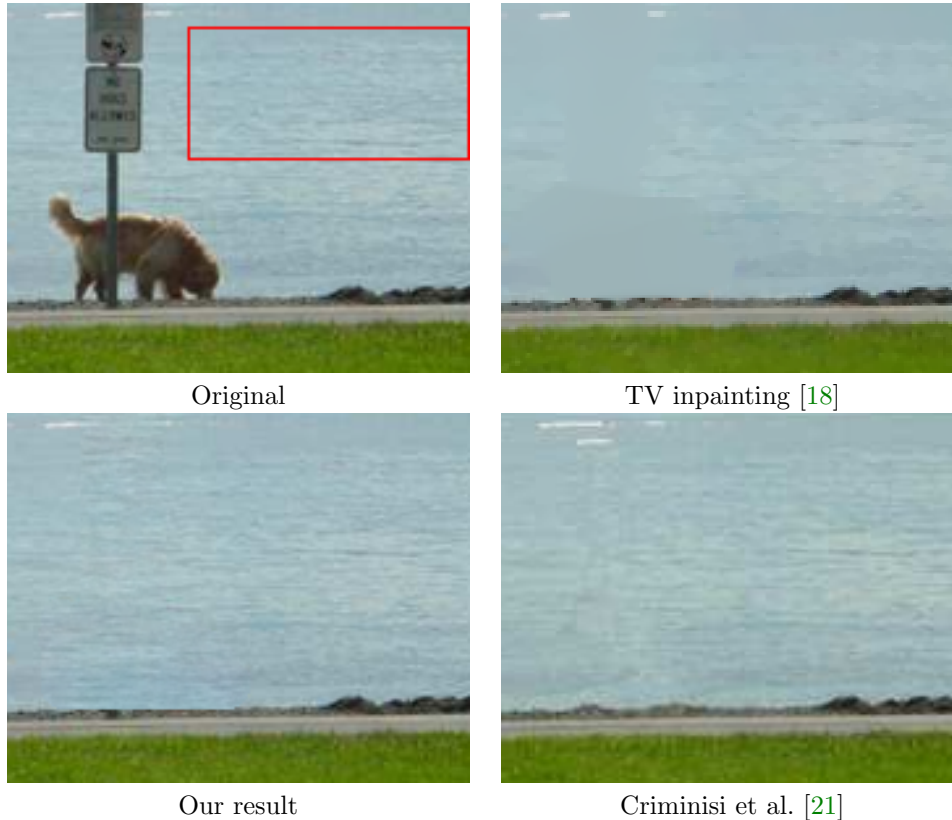


FIG. 13. *Comparison with [18, 21]. In the first row, we display the original image (taken from [21]) on the left, and on the right the result of TV inpainting [18] (obtained with the implementation available at [35]). In the second row, on the left we show the result of Gaussian inpainting (with a model estimated in the red box), and on the right the result of the patch-based method of [21]. As one can see, the TV inpainting is not able to preserve texture. In contrast, the method of [21] is truly able to generate textural content, but may lead to repetition artifacts.*

668 Gaussian inpainting gives nearly perfect results on microtextures (which could be
 669 expected since the Gaussian model is well adapted to such content). Also, the last
 670 rows of Fig. 14 show that the results obtained on macrotextures, although not perfect,
 671 are still quite convincing in comparison to patch-based methods. Even if Gaussian
 672 inpainting is not able to preserve salient geometric features, it has two important
 673 benefits: the synthesized content is smoothly blended in the input data, and the
 674 synthesized content does not suffer from repetition artifacts. But of course, Gaussian
 675 inpainting will clearly fail if one tries to inpaint a very thick hole in a highly non
 676 Gaussian texture (because the human visual system is able to discriminate between
 677 a highly structured texture and its ADSN counterpart).

678 All these remarks are confirmed with the results of Fig. 15 which provides a
 679 comparison of these methods on a difficult textural inpainting problem. This striking
 680 example clearly exhibits the benefits and drawbacks of each method. With Gaussian
 681 inpainting, the color distribution and frequential content are precisely respected, and
 682 long-range correlations are preserved (as can be seen in the kriging component), but
 683 complex geometric structures are not properly synthesized as they would be with
 684 a patch-based method. In contrast, with patch-based methods, we observe some

685 repetition artifacts which can be explained in the same way as the *growing garbage*
 686 effect which was already brought up by the seminal paper [27]. There may also be
 687 other artifacts which are more specific: on the result of [3], the inpainted domain is a
 688 bit too blurry and the border of the inpainted domain is still clearly visible; and on
 689 the result of [55], after close examination of the inpainted domain, we can perceive
 690 small seams which are due to changes in the offsets used for region pasting.

691 **5. Conclusion.** In this paper, we proposed a stochastic inpainting method based
 692 on Gaussian conditional simulation. It is able to inpaint holes of any shape and size
 693 in microtexture images while precisely respecting a random texture model. Gaussian
 694 texture inpainting shares of course some limitations with Gaussian texture synthesis,
 695 but we have illustrated on many texture images that this simple approach competes
 696 with state-of-the-art inpainting algorithms in terms of visual results (but not in com-
 697 putational time).

698 As discussed in the paper, we have proposed a very simple procedure for esti-
 699 mating a Gaussian texture model from a masked exemplar texture. Numerical ex-
 700 periments show that this naive technique gives good results provided that the mask
 701 complement contains a sufficiently plain piece of texture. Still, we believe that it
 702 would be interesting to dispose of a more robust estimation technique amenable to
 703 deal with very irregular masks (which amounts to estimating the texture covariance
 704 from sparsely sampled data). This may be rephrased as parameter estimation with
 705 hidden variables and might be addressed with an expectation-maximization technique,
 706 but keeping the computational cost of such a procedure seems very challenging.

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

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

717 We thank Alasdair Newson for his comments and for providing us with the im-
 718 plementation of [55]. We also thank Olivier le Meur, Lionel Moisan, and Frédéric
 719 Richard for several discussions about inpainting or kriging estimation.

720

REFERENCES

- 721 [1] A. ALMANSA, F. CAO, Y. GOUSSEAU, AND B. ROUGÉ, *Interpolation of digital elevation models*
 722 *using AMLE and related methods*, IEEE transactions on geoscience and remote sensing,
 723 40 (2002), pp. 314–325.
- 724 [2] P. ARIAS, V. CASELLES, AND G. FACCILOLO, *Analysis of a Variational Framework for Exemplar-*
 725 *Based Image Inpainting*, Multiscale Modeling & Simulation, 10 (2012), pp. 473–514, doi:10.
 726 1137/110848281, <http://epubs.siam.org/doi/abs/10.1137/110848281>.
- 727 [3] P. ARIAS, G. FACCILOLO, V. CASELLES, AND G. SAPIRO, *A variational framework for exemplar-*
 728 *based image inpainting*, International Journal of Computer Vision, 93 (2011), pp. 319–347.
- 729 [4] J. AUJOL, S. LADJAL, AND S. MASNOU, *Exemplar-based inpainting from a variational point of*
 730 *view*, SIAM Journal on Mathematical Analysis, 42 (2010), pp. 1246–1285.
- 731 [5] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO, AND J. VERDERA, *Filling-in by joint*
 732 *interpolation of vector fields and graylevels*, IEEE Transactions on Image Processing, 10
 733 (2001), pp. 1200–1211.



FIG. 14. *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 [55], the result of variational non-local inpainting [3] (obtained with the online implementation of [30] using the NLmeans option), and the result of [15] (obtained with the publicly available G'MIC plugin for GIMP [65]). 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 macrot textures, 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.

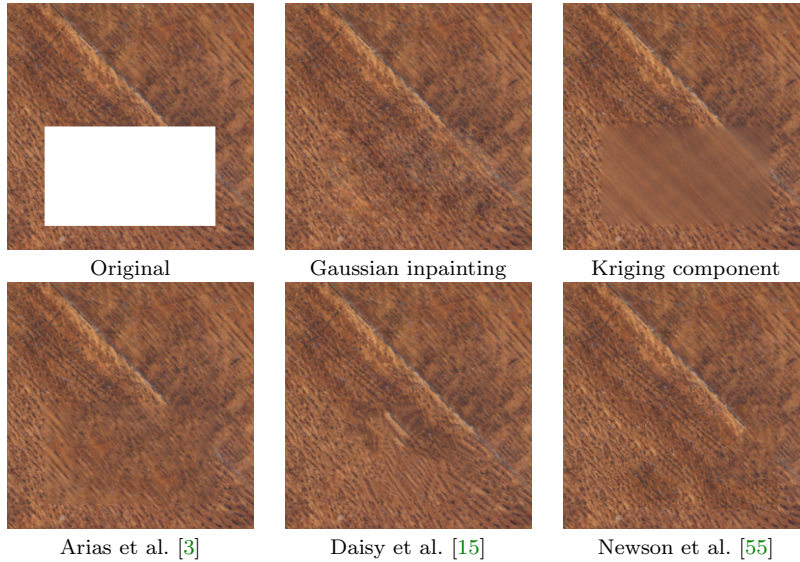


FIG. 15. *Comparison with patch-based methods (II)*. We compare several inpainting methods on a difficult textural inpainting problem. On the first row, from left or right, we display the masked input, the result of our method, together with the corresponding kriging component. On the second row, we display the results of variational non-local inpainting [3] (obtained with the online implementation of [30] using the NLmeans option), the result of [15] (obtained with the publicly available G'MIC plugin for GIMP [65]), and the result of [55]. 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 to state-of-the-art patch-based methods: with [3] the inpainted content is too blurry; with [15] we get repetition artifacts; and with [55] we can perceive small seams between inpainted regions using different offsets.

- 734 [6] C. BARNES, E. SHECHTMAN, A. FINKELSTEIN, AND D. GOLDMAN, *PatchMatch: a randomized*
735 *correspondence algorithm for structural image editing*, ACM Transactions on Graphics, 28
736 (2009).
- 737 [7] M. BERTALMIO, A. BERTOZZI, AND G. SAPIRO, *Navier-stokes, fluid dynamics, and image and*
738 *video inpainting*, in Proceedings of CVPR, vol. 1, IEEE, 2001.
- 739 [8] M. BERTALMIO, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image Inpainting*, in Proc. of
740 SIGGRAPH, 2000, pp. 417–424, doi:10.1145/344779.344972.
- 741 [9] M. BERTALMIO, L. VESE, G. SAPIRO, AND S. OSHER, *Simultaneous structure and texture image*
742 *inpainting*, IEEE Transactions on Image Processing, 12 (2003), pp. 882–889.
- 743 [10] A. BERTOZZI, S. ESEDOGLU, AND A. GILLETTE, *Inpainting of binary images using the Cahn-*
744 *Hilliard equation*, IEEE Transactions on image processing, 16 (2007), pp. 285–291.
- 745 [11] R. BORNARD, E. LECAN, L. LABORELLI, AND J. CHENOT, *Missing data correction in still im-*
746 *ages and image sequences*, in Proceedings of the tenth ACM international conference on
747 Multimedia, 2002, pp. 355–361.
- 748 [12] F. BORNEMANN AND T. MÁRZ, *Fast image inpainting based on coherence transport*, Journal of
749 Mathematical Imaging and Vision, 28 (2007), pp. 259–278.
- 750 [13] A. BUGEAU, M. BERTALMO, V. CASELLES, AND G. SAPIRO, *A comprehensive framework for*
751 *image inpainting*, IEEE Transactions on Image Processing, 19 (2010), pp. 2634–2645.
- 752 [14] M. BURGER, L. HE, AND C. SCHÖNLIEB, *Cahn-Hilliard inpainting and a generalization for*
753 *grayvalue images*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 1129–1167.
- 754 [15] P. BUYSSENS, M. DAISY, D. TSCHUMPERLÉ, AND O. LÉZORAY, *Exemplar-based Inpainting:*
755 *Technical Review and new Heuristics for better Geometric Reconstructions*, IEEE Trans-
756 actions on Image Processing, 24 (2015), pp. 1809–1824.
- 757 [16] J. CAI, R. CHAN, AND Z. SHEN, *A framelet-based image inpainting algorithm*, Applied and
758 Computational Harmonic Analysis, 24 (2008), pp. 131–149.
- 759 [17] F. CAO, Y. GOUSSEAU, S. MASNOU, AND P. PÉREZ, *Geometrically guided exemplar-based in-*

- 760 *painting*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 1143–1179.
- 761 [18] T. CHAN AND J. SHEN, *Mathematical models for local nontexture inpaintings*, SIAM Journal
762 on Applied Mathematics, 62 (2002), pp. 1019–1043.
- 763 [19] S. CHANDRA, M. PETROU, AND R. PIRODDI, *Texture Interpolation Using Ordinary Kriging*,
764 Pattern Recognition and Image Analysis, (2005), pp. 183–190.
- 765 [20] J.-P. CHILÈS AND P. DELFINER, *Geostatistics: modeling spatial uncertainty*, John Wiley &
766 Sons, 2009.
- 767 [21] A. CRIMINISI, P. PÉREZ, AND K. TOYAMA, *Region filling and object removal by exemplar-based*
768 *image inpainting*, IEEE Transactions on Image Processing, 13 (2004), pp. 1200–1212.
- 769 [22] L. DEMANET, B. SONG, AND T. CHAN, *Image inpainting by correspondence maps: a determin-*
770 *istic approach*, Applied and Computational Mathematics, 1100 (2003), p. 99.
- 771 [23] A. DESOLNEUX, L. MOISAN, AND S. RONSIN, *A compact representation of random phase and*
772 *Gaussian textures*, in Proceedings of ICASSP, 2012, pp. 1381–1384.
- 773 [24] J. DOOB, *Stochastic processes*, Wiley, 1990.
- 774 [25] D. C. DOWSON AND B. V. LANDAU, *The Fréchet distance between multivariate normal distribu-*
775 *tions*, Journal of Multivariate Analysis, 12 (1982), pp. 450–455, doi:10.1016/0047-259X(82)
776 90077-X.
- 777 [26] I. DRORI, D. COHEN-OR, AND H. YESHURUN, *Fragment-based image completion*, in ACM Trans-
778 actions on Graphics, vol. 22, 2003, pp. 303–312.
- 779 [27] A. A. EFROS AND T. K. LEUNG, *Texture synthesis by non-parametric sampling*, in Proceedings
780 of ICCV, vol. 2, 1999, pp. 1033–1038.
- 781 [28] M. ELAD, J. L. STARCK, P. QUERRE, AND D. L. DONOHO, *Simultaneous cartoon and texture*
782 *image inpainting using morphological component analysis (MCA)*, Applied and Computa-
783 tional Harmonic Analysis, 19 (2005), pp. 340–358, doi:10.1016/j.acha.2005.03.005.
- 784 [29] S. ESEDOGLU AND J. SHEN, *Digital inpainting based on the Mumford-Shah-Euler image model*,
785 European Journal of Applied Mathematics, 13 (2002), pp. 353–370.
- 786 [30] V. FEDOROV, G. FACCIOLO, AND P. ARIAS, *Variational Framework for Non-Local Inpainting*,
787 Image Processing On Line, 5 (2015), pp. 362–386, doi:10.5201/ipol.2015.136.
- 788 [31] B. GALERNE, Y. GOUSSEAU, AND J.-M. MOREL, *Random Phase Textures: Theory and Syn-*
789 *thesis*, IEEE Transactions on Image Processing, 20 (2011), pp. 257–267, doi:10.1109/TIP.
790 2010.2052822.
- 791 [32] B. GALERNE, A. LECLAIRE, AND L. MOISAN, *Texton Noise*. submitted to Computer Graphics
792 Forum.
- 793 [33] B. GALERNE, A. LECLAIRE, AND L. MOISAN, *A Texton for Fast and Flexible Gaussian Texture*
794 *Synthesis*, in Proceedings of EUSIPCO, 2014, pp. 1686–1690.
- 795 [34] B. GALERNE, A. LECLAIRE, AND L. MOISAN, *Microtexture Inpainting Through Gaussian Con-*
796 *ditional Simulation*, in Proceedings of the International Conference on Acoustics, Speech
797 and Signal Processing (ICASSP), IEEE, 2016.
- 798 [35] P. GETREUER, *Total Variation Inpainting using Split Bregman*, Image Processing On Line, 2
799 (2012), pp. 147–157, doi:10.5201/ipol.2012.g-tvi.
- 800 [36] C. GUILLEMOT AND O. LE MEUR, *Image inpainting: Overview and recent advances*, IEEE
801 Signal Processing Magazine, 31 (2014), pp. 127–144.
- 802 [37] K. HE AND J. SUN, *Image completion approaches using the statistics of similar patches*, IEEE
803 Transactions on Pattern Analysis and Machine Intelligence, 36 (2014), pp. 2423–2435.
- 804 [38] D. HEEGER AND J. BERGEN, *Pyramid-based texture analysis/synthesis*, in Proceedings of the
805 22nd annual conference on Computer graphics and interactive techniques, 1995, pp. 229–
806 238.
- 807 [39] Y. HOFFMAN AND E. RIBAK, *Constrained realizations of Gaussian fields: a simple algorithm*,
808 The Astrophysical Journal, 380 (1991), pp. L5–L8.
- 809 [40] H. IGEHY AND L. PEREIRA, *Image replacement through texture synthesis*, in Proceedings of
810 ICIP, vol. 3, 1997, pp. 186–189.
- 811 [41] J. JIA AND C. TANG, *Inference of segmented color and texture description by tensor voting*,
812 IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 771–786,
813 doi:10.1109/TPAMI.2004.10.
- 814 [42] W. KAMMERER AND M. NASHEED, *On the Convergence of the Conjugate Gradient Method*
815 *for Singular Linear Operator Equations*, SIAM Journal on Numerical Analysis, 9 (1972),
816 pp. 165–181, doi:10.1137/0709016.
- 817 [43] N. KOMODAKIS AND G. TZIRITAS, *Image completion using efficient belief propagation via priori-*
818 *ty scheduling and dynamic pruning*, IEEE Transactions on Image Processing, 16 (2007),
819 pp. 2649–2661.
- 820 [44] C. LANTUÉJOU, *Geostatistical Simulation: Models and Algorithms*, Springer, 2002.
- 821 [45] O. LE MEUR AND C. GUILLEMOT, *Super-resolution-based inpainting*, in ECCV 2012, Springer,

- 822 2012, pp. 554–567.
- 823 [46] A. LECLAIRE, *Random Phase Fields and Gaussian Fields for Image Sharpness Assessment and*
824 *Fast Texture Synthesis*, PhD thesis, Université Paris Descartes, 2015.
- 825 [47] J. LEWIS, *Texture Synthesis for Digital Painting*, in Proceedings of SIGGRAPH, 1984, pp. 245–
826 252, doi:10.1145/800031.808605.
- 827 [48] J. LEWIS, *Methods for Stochastic Spectral Synthesis*, in Proceedings on Graphics Interface,
828 1986, pp. 173–179.
- 829 [49] L. LI, T. ROMARY, AND J. CAERS, *Universal kriging with training images*, Spatial Statistics,
830 (2015).
- 831 [50] Y. LIU AND V. CASELLES, *Exemplar-based image inpainting using multiscale graph cuts*, IEEE
832 Transactions on Image Processing, 22 (2013), pp. 1699–1711.
- 833 [51] J. MAIRAL, M. ELAD, AND G. SAPIRO, *Sparse Representation for Color Image Restoration*,
834 IEEE Transactions on Image Processing, 17 (2008), pp. 53–69, doi:10.1109/TIP.2007.
835 911828.
- 836 [52] G. MARIETHOZ AND S. LEFEBVRE, *Bridges between multiple-point geostatistics and texture*
837 *synthesis: Review and guidelines for future research*, Computers & Geosciences, 66 (2014),
838 pp. 66–80.
- 839 [53] S. MASNOU, *Disocclusion: a variational approach using level lines*, IEEE Transactions on Image
840 Processing, 11 (2002), pp. 68–76.
- 841 [54] S. MASNOU AND J.-M. MOREL, *Level lines based disocclusion*, in Proceedings of ICIP, vol. 3,
842 1998, pp. 259–263, doi:10.1109/ICIP.1998.999016.
- 843 [55] A. NEWSON, A. ALMANSA, M. FRADET, Y. GOUSSEAU, AND P. PÉREZ, *Video Inpainting of*
844 *Complex Scenes*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1993–2019, doi:10.
845 1137/140954933.
- 846 [56] P. PÉREZ, M. GANGNET, AND A. BLAKE, *Patchworks: Example-based region tiling for image*
847 *editing*, Microsoft Research, MSR-TR-2004-04, Tech. Rep, (2004).
- 848 [57] G. PEYRÉ, *Texture Synthesis with Grouplets*, IEEE Transactions on PAMI, 32 (2010), pp. 733–
849 746, doi:10.1109/TPAMI.2009.54.
- 850 [58] L. RAAD, A. DESOLNEUX, AND J. MOREL, *A Conditional Multiscale Locally Gaussian Texture*
851 *Synthesis Algorithm*, Journal of Mathematical Imaging and Vision, (2016), pp. 1–20.
- 852 [59] L. RAAD, A. DESOLNEUX, AND J.-M. MOREL, *Conditional Gaussian Models for Texture Syn-*
853 *thesis*, in Proceedings of Scale Space and Variational Methods in Computer Vision, 2015.
- 854 [60] H. RUE, *Fast sampling of Gaussian Markov random fields*, Journal of the Royal Statistical
855 Society: Series B (Statistical Methodology), 63 (2001), pp. 325–338.
- 856 [61] H. RUE AND L. HELD, *Gaussian Markov Random Fields: Theory and Applications*, CRC Press,
857 2005.
- 858 [62] C. SCHÖNLIEB, *Partial Differential Equation Methods for Image Inpainting*, Cambridge Uni-
859 versity Press, 2015.
- 860 [63] J. SUN, L. YUAN, J. JIA, AND H. SHUM, *Image completion with structure propagation*, in ACM
861 Transactions on Graphics, vol. 24, ACM, 2005, pp. 861–868.
- 862 [64] A. TELEA, *An image inpainting technique based on the fast marching method*, Journal of Graph-
863 ics Tools, 9 (2004), pp. 23–34.
- 864 [65] D. TSCHUMPERLÉ, *GREYC’s Magic for Image Computing (GIMP plugin)*, <http://gmic.eu/>.
- 865 [66] J. J. VAN WIJK, *Spot noise texture synthesis for data visualization*, in Proc. of SIGGRAPH,
866 vol. 25, 1991, pp. 309–318.
- 867 [67] L. WEI AND M. LEVOY, *Fast texture synthesis using tree-structured vector quantization*, in
868 Proceedings of SIGGRAPH, 2000, pp. 479–488.
- 869 [68] Y. WEXLER, E. SHECHTMAN, AND M. IRANI, *Space-Time Completion of Video*, IEEE Trans-
870 actions on Pattern Analysis and Machine Intelligence, 29 (2007), pp. 463–476, doi:10.1109/
871 TPAMI.2007.60.
- 872 [69] G. XIA, S. FERRADANS, G. PEYRÉ, AND J. AUJOL, *Synthesizing and Mixing Stationary*
873 *Gaussian Texture Models*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 476–508,
874 doi:10.1137/130918010.
- 875 [70] Z. XU AND J. SUN, *Image inpainting by patch propagation using patch sparsity*, IEEE Trans-
876 actions on Image Processing, 19 (2010), pp. 1153–1165.