A Texture Synthesis Model Based on Semi-discrete Optimal Transport in Patch Space

Bruno Galerne, Arthur Leclaire, Julien Rabin

A TEXTURE SYNTHESIS MODEL BASED ON SEMI-DICRETE OPTIMAL TRANSPORT IN PATCH SPACE

B. GALERNE *, A. LECLAIRE†, AND J. RABIN ‡

Abstract. Exemplar-based texture synthesis consists in producing new synthetic images which have the same perceptual characteristics than a given texture sample while exhibiting sufficient innovation (to avoid verbatim copy). In this paper, we propose to address this problem with a model obtained as local transformations of Gaussian random fields. The local transformations operate on 3×3 patches and are designed to solve a semi-discrete optimal transport problem in order to reimpose the patch distribution of the exemplar texture. The semi-discrete optimal transport problem is solved with a stochastic gradient algorithm, whose convergence speed is evaluated on several practical transport cases.

After studying the properties of such transformed Gaussian random fields, we propose a multiscale extension of the model which aims at preserving the patch distribution of the exemplar texture at multiple scales. Experiments demonstrate that this multiscale model is able to synthesize structured textures while keeping several mathematical guarantees, and with low requirements in synthesis time and memory storage. In particular, a single patch optimal transport map is shown to be better than iterated nearest neighbor assignments in terms of statistical guarantees. Besides, once the model is estimated, the resulting synthesis algorithm is fast and highly parallel since it amounts to perform weighted nearest neighbor patch assignments at each scale.

Key words. optimal transport, texture synthesis, patch distribution, nearest neighbor assignments

AMS subject classifications. 62M40, 65D18, 65K10, 68U10,

1. Introduction. [1]

In computer graphics or film rendering, it is often desirable to cover objects with detail patterns that look like natural textures. For that purpose, there is a need for algorithms that take a sample of a natural texture as input and are able to produce a (possibly much larger) new texture image which has the same perceptual characteristics. This problem, called exemplar-based texture synthesis, is by nature ill-posed, but one can set up some ideal guidelines to answer it properly. For example: the output texture should everywhere locally resemble to one part (at least) of the input; nonetheless the output must exhibit some innovation with respect to the input, meaning that verbatim copy is not an acceptable solution, *etc.* And beyond the quality of the synthesized image, some features of the model may suit given technical constraints depending on the application. For instance, one may require fast and parallel algorithms for the purpose of real-time image synthesis.

In order to study this problem, researchers have first considered the framework of homogeneous textures, *i.e.* texture images which are statistically translation invariant. Although quite restrictive (this class is not invariant to rough changes in illumination or viewpoint), this is a convenient framework to model a texture as a realization of a stationary random field. In this framework, texture analysis consists in inferring the distribution of this random field from one texture sample, while texture synthesis can be understood as drawing a sample of this model. But the difficult question is now to wisely choose stochastic models whose parameters can be inferred

*Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité, France. (Bruno.Galerne@parisdescartes.fr).

†CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235, Cachan, France. (Arthur.Leclaire@cmla.ens-cachan.fr).

‡Normandie Univ, ENSICAEN, UNICAEN, CNRS, GREYC, 14050 Caen, France. (Julien.Rabin@unicaen.fr).

[1] A preliminary version of this work was published as a conference paper in [17].

from one realization, and which can be sampled efficiently. In relation to this issue lies the question of statistical guarantee: during this analysis-synthesis pipeline, which statistics would we like to preserve with a precise control?

From the psychovisual perspective, B. Julesz conjectured in [25, 26] that all first and second-order statistics of the model are sufficient to characterize a texture, for example the mean color, the correlations between pixel values, the density of a (possibly non-linear) filter response, correlations between filter responses, and so on. Even if he designed counter-examples that refute the first versions of his conjecture, the initial assertion still holds for a very large class of textures. This work suggests the following flexible methodology for texture synthesis: first, choose a set of statistics that are relevant for texture perception; and second, optimize the synthesized image to match these statistics with the ones observed on the exemplar texture. The algorithms following this methodology are sometimes referred to as parametric synthesis methods.

One instance is given by the method of Heeger and Bergen [24] which imposes the density of subband responses to a steerable pyramid. Another one was designed by Portilla and Simoncelli [50] which is based on first and second-order moments, in particular correlations in a complex wavelet transform, making use of a gradient descent based algorithm. One breakthrough of Portilla and Simoncelli was to propose a set of 710 statistical measures that are sufficient to characterize a very large class of texture images, on the basis of a precise perceptual study. More recently, Gatys et al. [18] suggested to use a similar algorithm with responses to a pre-trained convolutional neural network. More precisely they extract spatially averaged, non-centered correlations (Gram matrices) at certain layers. An important difference between [50] and [18] is that the latter method uses much more parameters (≈ 170000) whose psychovisual interpretation is not *a priori* justified. But at this cost, the method of [18] is able to synthesize nearly every real-life texture in a quite convincing way. Notice also that a similar methodology, but based on first-order moments of scattering coefficients, was used in [4] for audio texture synthesis. The work of Gatys et al [18] was followed by several variants that improve the quality of synthesis or the computational time. Liu et al. [40] propose to include a spectrum constraint in the framework of [18] to better preserve the frequency content of the input. Berger and Memisevic [3] and Sendik and Cohen-Or [57] propose to include spatial shifts in the Gram matrices for synthesis of structured textures. Ulyanov et al. [61] propose to estimate a feed-forward convolutional network that mimics the optimization procedure of [18] in order to address real-time texture synthesis.

These statistically inspired methods define a texture model which can only be expressed as the limit (should it exist) of the optimization procedure, starting from a white noise. In parallel, several authors have considered texture models that are inherently designed to respect a statistical constraint in average (not exact matching) while satisfying a maximal entropy principle. This point of view was proposed by Zhu, Wu and Mumford [69] who defined the FRAME model (Filters, Random fields, And Maximum Entropy). They proposed a methodology based on stochastic optimization (and Gibbs sampling) to estimate and sample a FRAME model. Unfortunately, such Monte-Carlo methods are very slow, and thus this framework is restricted to very quantized textures and very few filters. A variant of FRAME, proposed in [41], copes with some of these issues by replacing the Gibbs sampler with a Langevin diffusion, at the cost of loosing guarantee of convergence. Generally speaking, as discussed by the authors of [4], the parametric methods like [50] that perform gradient descent with white noise initialization can be considered as a very rough approximation of a

maximum entropy model.

Going back in time, as a simple case of maximum entropy models, stationary Gaussian random fields are the maximum-entropy stationary models which respect the first and second-order moments of the texture, *i.e.* the mean color value and the power spectrum. These Gaussian models were actually used (under different names like convolution noise or asymptotic spot noise) in early texture models in computer graphics [34, 35, 62] and later studied in more details for exemplar-based synthesis in [14, 16]. Such Gaussian random fields are actually the perfect solutions for second-order modeling because they are of maximal entropy among stationary models with prescribed second-order moments, and besides they allow for efficient inference or sampling.

As explained in [69, 45], the solution to the maximum entropy problem can always be expressed as a Gibbs distribution. Gibbs distributions, *a.k.a.* Markov random fields (MRF) were actually encountered in early image and texture modeling [9, 5, 8, 6, 19, 10, 7]. The use of MRF models for texture is motivated by the fact that texture perception is governed by the local interactions between the pixel values, which can be encoded in the local conditional distribution (which defines the Gibbs distribution thanks to Hammersley-Clifford theorem). These early MRF texture models were characterized by very few parameters that define the local conditional distribution. In all these works, the estimation of parameters relies on maximum likelihood, which requires in general an iterative procedure. Let us mention that the stochastic optimization procedure adopted in [19] for estimating parameters of MRF models is actually the same as the one proposed in [69], and was already studied by Lippman in [39].

These parametric MRF models had quite limited expressive range because of the too small number of parameters. Paget and Longstaff [46] lifted this restriction with a nonparametric estimation of the local conditional distribution. As attested by the visual results shown in [46], their model is thus much richer than previous parametric MRF models, but the corresponding algorithm is yet quite slow. From this first nonparametric model, a huge progress for synthesis has been made by Efros and Leung [13] who proposed to replace the complex nonparametric estimate of the local conditional distribution by a simple empirical distribution with neighborhood constraints.

The seminal work by Efros and Leung [13] has paved the way for many successful synthesis methods that operate on patches (*i.e.* on small square subimages of fixed size). Broadly speaking, the first patch-based methods (like the one of [13]) consist in progressive filling of the synthesis domain by (randomized) local copy-paste operations depending on the current surrounding patch. Wei and Levoy [66] proposed a multiscale variant of Efros-Leung method and replaced the local resampling by a deterministic projection on the exemplar patches; they also proposed an acceleration of the nearest neighbor search based on tree-structured vector quantization. Liang et al. [38] proposed to accelerate the Efros-Leung method by copying whole patches instead of processing pixels one by one. Efros and Freeman built on this method and carefully handled the boundaries between the patches: they proposed image quilting [12] to compute a minimum error seam between adjacent copied blocks. This method was later accelerated by Kwatra et al. [29] who applied the graphcut method to compute the minimum error cut between two adjacent patches. More recently, Raad et al. [52] designed a kind of randomized multiscale version of image quilting where each new patch is drawn from a local Gaussian model (which is estimated from a set of similar patches, computed with a distance depending on the current synthesis

at the same scale and at the adjacent coarser scale). Also, Li and Wand [36, 37] addressed image and texture synthesis by considering "neural patches", that is, patches extracted from layers of a convolutional neural network. Apart from these works, many other authors have contributed to the field of patch-based methods, and the reader is referred to [65, 51] for a more exhaustive overview of the state of the art. One common drawback of patch-based methods is the lack of statistical control. Although Levina and Bickel [32] showed the consistency of the Efros-Leung resampling method, their result only holds true in an asymptotic framework where the input size tends to infinity. In practice, as mentioned in [13, 1], one can observe that the progressive filling of the pixels may get stuck in a local neighborhood of the input, thus repeating a small part of the exemplar in an absurd manner; we then say that the algorithm starts to *grow garbage*.

Patch-based texture synthesis has later inspired texture optimization which was first proposed by Kwatra et al. [28]. It consists in iterative minimization of a functional that encodes the similarity at multiple scales between the patches of the synthesis and the patches of the example, starting from a white noise at coarse scale. The rationale behind this model is that every patch of the synthesis should resemble at least one patch of the exemplar texture, and it should be so at several scales. One advantage of such model compared to [13] for instance, is that it formulates a global image model which does not depend on any pixel-filling order. Therefore, the *growing garbage* effect is attenuated in such global model. However, as will be discussed later in the present paper, solving such a patch-based optimization problem does not directly provide a statistical control on the output texture. Indeed, the fact that every part of the output should be encountered in the input does not ensure that it is encountered in the same proportion (thus providing no symmetric guarantee that every part of the input is encountered in the output). Nevertheless, the texture optimization method of Kwatra et al. inspired the (very fast) parallel controllable texture synthesis method of Lefebvre and Hoppe [31] and several variants [23, 22, 11].

The methodology of Kwatra et al. [28] was later generalized to other functionals, thus opening a field which is now referred to as variational texture synthesis. It can be thought of as a generalization of parametric texture synthesis: the corresponding functionals contain, in addition to statistically-inspired terms, some measure of regularity (*e.g.* related to the local sparsity in a visual dictionary). Following a first model of Peyré [47] that exploits the sparsity of patches in an adapted dictionary, Tartavel et al. [59] proposed to minimize a functional that combines three terms: a Wasserstein distance used to compare the color distributions, a frequency term that compares the power spectra, and a third term which is related to the sparsity of patch decompositions in an adapted dictionary.

Before explaining our contribution, let us remark that the interface between parametric and non-parametric texture models seems a bit porous. For example, the method of Heeger and Bergen is often categorized in parametric texture synthesis but it imposes the p.d.f. for each subband, and not only a finite set of statistics. Instead of parametric methods, it seems wiser to make a distinction between statistically inspired methods and patch-based methods as is done in [51]. But either this does not seem fair to patch-based methods which, as argued in the seminal paper [13], are originally inspired by a non-parametric estimation of the local conditional distribution [48]. To summarize, the success of all texture synthesis methods relies on the two following goals (which are, more or less, directly or indirectly, achieved)

1. extraction of local features in a translation invariant manner that is correlated with human perception, for example with color attributes, Fourier coefficients,

filter responses, patch attributes, textons (in the sense of Julesz [26]),
2. global statistical control.

As we already said, texture perception is governed by local interactions between pixels. Although motivated by the popularity of early texture MRF models, one can very well question the use of local *conditional* distributions for encoding the local interactions. In other words, is the conditional aspect really important, and why wouldn't we consider directly the patch distribution? This question was discussed by Varma and Zissermann in [63]. They first show that for texture classification, defining texture classes using the distribution of raw pixel intensities in small patches (3×3 or 5×5) can achieve better results than previous approaches based on filter responses (with larger support filters). They also exhibit a gain when classifying using local conditional distributions, but the gain is small and this procedure naturally entails some difficulties in the estimation. Actually, as soon as we extract enough independent 3×3 filter responses, then preserving the joint distribution of these responses is exactly equivalent to preserving the distribution of 3×3 patches. Thus, for some applications where filtering does not *a priori* simplify the problem, one may very well work directly on the patch distribution.

In this work, we propose to address texture synthesis by acting directly on the patch distribution at multiple scales using adapted local transformations. The link between local transformations and global statistical control is made possible by using semi-discrete optimal transportation. Optimal transport (OT) consists in computing measurable mappings which send one probability distribution onto another one while minimizing a transportation cost. We shall speak of discrete OT if both source and target measures are discrete, continuous OT if they are continuous, and semi-discrete OT if the source measure is continuous while the target measure is discrete. OT has already been used in the past to address several image processing problems. For example Rabin et al. proposed to consider Wasserstein distances to compare histograms of gradient orientations for matching image features [53]. The same authors made connections between color transfer and optimal transport in the color space [54]. Wasserstein barycenters were used in [55, 67] to address texture mixing. Tartavel et al. [60] extended variational texture synthesis by combining discrete OT distances computed on several (non-linear) filter responses. Finally, Gutierrez et al. [21] proposed a texture synthesis method that enforces the patch distribution at multiple scales by applying a discrete OT plan. This method can be understood as an elaborate improvement of the texture optimization method [28] with a global statistical control. But its main drawback is that it is quite slow due to the computations of discrete OT plans, and thus it is not scalable for synthesis of large images.

Here we will design a multiscale texture model that relies on semi-discrete optimal transportation to reimpose the patch distribution at multiple scales. In the paper [2] about least square clustering, Aurenhammer et al. showed that a solution to semi-discrete OT could be found in the form of weighted nearest neighbor (NN) assignments. In order to be optimal, the weights defining the transport maps should solve a $\mathcal{C}^1$ concave maximization problem. Several gradient-based schemes have been proposed to solve this optimization problem [43, 33] and in particular a provably convergent damped Newton algorithm in [27]. But these methods require exact computation of measures of polytopes, which is difficult in high dimension. Here, we will exploit and study a stochastic optimization algorithm for semi-discrete OT that was recently proposed in [20]. Even if this average stochastic gradient descent (ASGD) is quite slow, it provides a good approximation to semi-discrete OT in patch space (in dimension 27 for 3×3 color patches).

Starting from an adapted Gaussian noise at coarse scale, the proposed model consists in applying weighted NN assignments in order to reimpose the 3×3 patch distribution at each scale, and going from one scale to the next one with exemplar-based upsampling. Heuristically speaking, the Gaussian model of the first layer sets the medium-range correlations of the texture whereas the further patch transformations add geometric details in a statistically coherent manner. Therefore, our algorithm is yet another bridge between the parametric and non-parametric synthesis models. The model estimation requires one pass of synthesis during which, at each scale, a source distribution (Gaussian mixture model) and an OT map are estimated. Once estimated, the OT maps are stored, and can be used for the synthesis of possibly very large images. Our method can be seen as a localized version of [21]. Indeed, in contrast to [21], our model leads to a much faster (and highly parallel) synthesis algorithm because the local patch transformations are computed once and for all, and are applied independently to all patches at each scale.

Compared to other famous texture models, we will see that this multiscale OT-based model produces visual results that are close to the state-of-the-art with a low computational time and memory storage. In particular, compared to models based on iterated NN assignments [28], this model provides a more precise global statistical control, both on the medium-range correlations and on the patch distribution. In other words, only one OT map can do better than iterated NN projections, while being much faster. Compared to variational texture models, the multiscale OT model is much lighter and highly parallel, thus able to synthesize very large images in a few seconds. The computational time is better than the first neural-based texture models [18]; however the class of well synthesized textures is smaller because OT maps are not able to retrieve strong local geometric constraints, and because a small quantity of blur is induced by the patch recomposition strategy (by ℓ^2 average). Yet, we will provide several examples for which the OT-based model performs better than [18] thanks to the global statistical control. Another benefit of this model is that it has essentially two parameters (number of scales and number of components in the GMM patch distributions) that can be easily tuned manually.

The paper is organized as follows. In Section 2 we recall the framework of semi-discrete optimal transport and propose several practical cases to examine the convergence of the stochastic gradient scheme. In Section 3 we introduce the OT-based texture model, beginning with the monoscale model formulated as a local transform of a Gaussian random field, and next the multiscale extension. In Section 4, we present many texture synthesis results both for the monoscale and multiscale models, which demonstrate the benefit of global statistical control on multiscale patch distributions. We conclude the paper in Section 5 by raising issues concerning geometric models for the patch space and the putative limitations of statistically-inspired texture synthesis.

2. Semi-discrete Optimal Transport.

In this section, we recall the framework for semi-discrete optimal transport established in [2, 27, 20]. Also we propose a detailed numerical study of a stochastic gradient algorithm used for solving this optimal transport problem.

2.1. The Optimal Transport Problem and its Dual Formulation.

Let μ, ν be two probability measures on $\mathbb{R}^D$. If T is a measurable map, we denote by $T_\sharp \mu$ the push-forward measure defined as $T_\sharp \mu(A) = \mu(T^{-1}(A))$. The Monge formulation of L^2 optimal transport from μ to ν is formulated as

$$\text{(OT-M)} \qquad \inf \int_{\mathbb{R}^D} \|x - T(x)\|^2 d\mu(x)$$

where the infimum is taken over all measurable maps $T : \mathbb{R}^D \to \mathbb{R}^D$ such that $T_\sharp \mu = \nu$. A convex relaxation of this problem is given by the Kantorovich formulation

$$\text{(OT-K)} \qquad \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^D \times \mathbb{R}^D} \|x-y\|^2 d\pi(x,y).$$

where $\Pi(\mu,\nu)$ is the set of probability measures on $\mathbb{R}^D \times \mathbb{R}^D$ having marginal distributions μ, ν. It is clear that (OT-M) can be seen as a restriction of (OT-K) for transport plans π of the form $(\mathrm{Id} \times T)_\sharp \mu$ (whose support is contained in the graph of T). General conditions for existence and unicity of solutions can be found in [64] and [56].

In this paper we will concentrate on the semi-discrete case, meaning that μ is an absolutely continuous distribution and that ν has a finite support. More precisely, in all the following, we assume that μ has a bounded probability density function ρ and that ν is a discrete measure $\nu = \sum_{y \in S} \nu_y \delta_y$ with finite support S. As proved in [2, 33, 27], taking the convex dual of (OT-K) leads to a finite-dimensional convex optimization problem. Somehow, this amounts to consider maps given by biased nearest neighbor assignments

$$T_v(x) = \operatorname*{argmin}_{y \in S} \|x-y\|^2 - v(y), \tag{1}$$

where $v \in \mathbb{R}^S$ is a finite set of scalars. This map T_v is defined almost everywhere, and its preimages define a partition of $\mathbb{R}^D$ up to a negligible set, called the power diagram (or also Laguerre tessellation)

$$\mathrm{Pow}_v(y) = \{\ x \in \mathbb{R}^D \mid \forall z \in S \setminus \{y\},\ \|x-y\|^2 - v(y) < \|x-z\|^2 - v(z)\ \}. \tag{2}$$

When $v = 0$, we get the nearest neighbor (NN) projection which assigns to x the closest point in S (unique for almost all x), whose preimages form the Voronoi diagram. Including the scalars $(v(y))_{y \in S}$ (called power weights) in the comparison allows to move the boundaries of the cells. Notice that some power cells may vanish for particular values of v. For convenience of notation, we will also define the c-transform of v with respect to the cost $c(x,y) = \|x-y\|^2$ as

$$v^c(x) = \min_{y \in S} \|x-y\|^2 - v(y). \tag{3}$$

In the semi-discrete setting, solving the transport problem mainly consists in splitting the source mass with a power diagram in such a way that the μ-measure of each power cell corresponds to the ν-measure of the associated point. This is summarized in the following theorem, which is recalled without proof.

THEOREM 1 ([2, 27]). *The semi-discrete optimal transport problem (OT-K) admits solutions of the form T_v where v solves the concave optimization problem*

$$\operatorname*{argmax}_{v \in \mathbb{R}^S} H(v) \quad \textit{where} \quad H(v) = \int_{\mathbb{R}^D} v^c(x) d\mu(x) + \sum_{y \in S} v(y) \nu_y. \tag{4}$$

Besides, the function H is $\mathcal{C}^1$-smooth and its gradient is given by

$$\frac{\partial H}{\partial v(y)} = -\int_{\mathrm{Pow}_v(y)} \rho(x) dx + \nu_y = -\mu(\mathrm{Pow}_v(y)) + \nu_y. \tag{5}$$

Thus, v is a critical point of H if and only if $\mu(\mathrm{Pow}_v(y)) = \nu_y\ \forall y$, i.e. $(T_v)_\sharp \mu = \nu$.

Remark (about unicity): There is actually an infinity of solutions v to the problem (4) because H is invariant to the addition of a constant. However, for the case we consider here (L^2 cost, μ absolutely continuous), under the assumptions $\int \|x\|^2 d\mu(x) < \infty$ (which is often satisfied in practice), the problem (OT-K) admits a unique solution [56]. Therefore, Theorem 1 provides the solution of (OT-M) which is uniquely defined for almost every x even if there exists several different v defining the same assignment T_v.

2.2. Stochastic Optimization. Thanks to Theorem 1, a solution T_v can be numerically computed using a gradient-based optimization of the function H and convergence will be guaranteed thanks to the concavity of H. But a difficulty of such approaches is that computing the gradient (5) amounts to computing the μ-measures of the power cells. In small dimensions (≤ 3) and for particular source measures μ, it is possible to numerically evaluate the gradient (and even the Hessian) with good precision, which has been exploited in quasi-Newton schemes in [43, 33, 27]. But in higher dimensions, it is harder to explicitly compute the geometry of the power cells, and thus exact gradient computations are not tractable.

In a high-dimensional setting, one may turn to using Monte-Carlo estimates for the gradient instead of exact computations. In other words, the maximization of H can be addressed with stochastic gradient ascent, which is made possible by writing

$$H(v) = \mathbb{E}[h(X, v)] \quad \text{where} \quad h(x, v) = v^c(x) + \sum_{y \in S} v(y)\nu_y \tag{6}$$

and where X is a random variable of distribution μ. Notice that for $x \in \mathrm{Pow}_v(y)$, $v \mapsto v^c(x)$ is smooth with gradient $-e_y$ (where (e_y) is the canonical basis of $\mathbb{R}^S$). Therefore, for any $w \in \mathbb{R}^S$, for almost all $x \in \mathbb{R}^D$, $v \mapsto h(x, v)$ is differentiable at w and

$$\nabla_v h(x, w) = -e_{T_w(x)} + \nu. \tag{7}$$

In order to minimize $-H$, Genevay et al. [20] recently proposed the following averaged stochastic gradient descent (ASGD) initialized with $\tilde{v}^1 = 0$

$$\begin{cases} \tilde{v}^k &= \tilde{v}^{k-1} + \frac{C}{\sqrt{k}} \nabla_v h(x^k, \tilde{v}^{k-1}) \quad \text{where } x^k \sim \mu \\ v^k &= \frac{1}{k}(\tilde{v}^1 + \ldots + \tilde{v}^k). \end{cases} \tag{8}$$

Since $\nabla_v h(x, \tilde{v}^{k-1})$ exists x-a.s. and is bounded, the convergence of this algorithm is ensured by [44, Th.7], in the sense $\max(H) - \mathbb{E}[H(v_k)] = \mathcal{O}(\frac{\log k}{\sqrt{k}})$. For the sake of completeness, we recall the proof of convergence in Appendix A. This proof is not affected by the fact that $h(\cdot, v)$ is only differentiable almost everywhere because μ is absolutely continuous.

2.3. Convergence study. This section is devoted to a numerical study of the stochastic algorithm (8) for semi-discrete optimal transport. The behavior of this algorithm is briefly discussed in the original conference paper [20]. Here we propose to examine the convergence speed on various practical cases where the optimal solution can be computed in closed form.

2.3.1. One-dimensional case study. In the one-dimensional case ($D = 1$), the semi-discrete optimal transport cost can be computed with a closed form formula based on cumulative distribution functions [64]. Indeed, let us consider the cumulative

distribution function Φ of the density ρ, $\Phi(x) := \int_{-\infty}^{x} \rho(t)dt$, and Φ^{-1} its generalized inverse defined by

$$\Phi^{-1}(a) = \inf\{\ x \in \mathbb{R} \mid \Phi(x) \geqslant a\ \} \in [-\infty, +\infty], \quad a \in [0,1]. \tag{9}$$

Let us assume for the sake of simplicity that the locations of the target Dirac masses are sorted in increasing order: $S = \{y_1 < y_2 < ... < y_J\}$. We then obtain an optimal assignment T by setting

$$T(x) = y_i \quad \forall x \in (x_{i-1}, x_i) \quad \forall 1 \leq i \leq J \tag{10}$$

where $(x_0 = -\infty, x_1, \ldots, x_{J-1}, x_J = +\infty)$ is any partition of $\mathbb{R}$ (up to a negligible set) into J intervals, the μ-measures of which equal the values ν_i:

$$\mu([x_{i-1}, x_i]) = \int_{x_{i-1}}^{x_i} \rho(x)dx = \nu(\{y_i\}) = \nu_i. \tag{11}$$

PROPOSITION 2. *In the one-dimensional case, one optimal assignment T_{v^*} is obtained by setting*

$$\forall\ 1 \leq i \leq J, \quad v_i^* = v_1^* + 2 \sum_{1 \leq j < i} (y_{j+1} - y_j) \left(-x_j^* + \frac{y_{j+1} + y_j}{2} \right), \tag{12}$$

where the boundaries x_i^* of the power cells $\mathrm{Pow}_{v^*}(y_i) = (x_{i-1}^*, x_i^*)$ are defined by

$$\forall i \in \{1, \ldots, J-1\}, \quad x_i^* = \Phi^{-1}\Big(\sum_{1 \leq j \leq i} \nu_j \Big) \tag{13}$$

and by convention $x_0^* = -\infty$ and $x_J^* = +\infty$.

Proof. First, let us notice that (x_i^*) is an increasing sequence since $\nu_i > 0$ for all i. Besides, since μ has a density, we have $\Phi(x_i^*) = \sum_{1 \leq j \leq i} \nu_j$ and thus the intervals $[x_{i-1}, x_i]$ satisfy (11). Thus we only have to propose a weight vector v for which the associated power cells are exactly these intervals. If v is such a vector, then for fixed $i \in \{1, \ldots, J-1\}$, we compute v_i by looking at the interface between power cells (x_{i-1}^*, x_i^*) and (x_i^*, x_{i+1}^*). Since they are both non-empty, coming back to the definition (2) we get at the interface the equality

$$|x_i^* - y_{i+1}|^2 - v_{i+1} = |x_i^* - y_i|^2 - v_i \tag{14}$$

$$\text{i.e.} \quad v_{i+1} = v_i + (y_{i+1} - y_i)(-2x_i^* + y_{i+1} + y_i). \tag{15}$$

We thus obtain (12) by trivial recursion. Conversely, one can check that defining v^* by (12) leads to the power cells $\mathrm{Pow}_{v^*}(y_i) = (x_{i-1}^*, x_i^*)$.

Remark (about unicity): This is not true in general that v^* is the unique solution of the problem and there are two reasons for that. The first (trivial) one is that the power cells are clearly invariant when adding a constant to all weights v_i. The second one is related to the fact that the L^2 optimal assignment is only unique μ-almost everywhere [64]. In particular, when the source distribution μ has a disconnected support, then several solutions can appear. For example, in the case

$$\mu(x) = \mathbf{1}_{[-1,-0.5] \cup [0.5,1]}(x)dx \quad , \quad \nu = 0.5(\delta_{-1} + \delta_1) \tag{16}$$

then T_v is optimal as soon as the boundary of the two power cells belongs to $[-0.5, 0.5]$. This reflects the fact that there may be several partitions $(x_0, x_1, \dots, x_J)$ satisfying (11).

Remark (about the link between x and v): In the following numerical study, we shall need to compute the power cells associated to a vector v (not necessarily the optimal one). In this case, we need a more complex recursion than (15). Indeed, in order to obtain the equality (14), we exploited the fact that for an optimal assignment, all power cells are non-empty and thus we can obtain x^* from v^* by the simple formula

$$\forall i \in \{1, \dots, J-1\}, \quad x_i^* = \frac{y_{i+1} + y_i}{2} - \frac{v_{i+1}^* - v_i^*}{2(y_{i+1} - y_i)}. \tag{17}$$

Now if v is any vector (not necessarily a solution to (4)), then some of the associated power cells may collapse, and thus this formula does not hold true anymore. However, one can still compute x from v in a recursive manner, as follows.

$x_0 = -\infty$, $i = 1$
For $j = 1, \dots, J-1$
  $x' \leftarrow \frac{y_{j+1}+y_i}{2} - \frac{v_{j+1}-v_i}{2(y_{j+1}-y_i)}$
  If $x' > x_{i-1}$ and $\mathrm{Pow}_v(y_{j+1}) \neq \varnothing$
    Then
      $\forall k = i, \dots, j, \quad x_k \leftarrow x'$
      $i \leftarrow j + 1$

In order to prove that this recursion holds, one should adapt the proof of Proposition 2 by handling the case when power cells may vanish. Notice that, once x' has been affected, it is possible to ensure that $\mathrm{Pow}_v(y_{j+1}) \neq \varnothing$ by checking if

$$(x' - y_{j+1})^2 - v_{j+1} < \min_{k \notin [i, j+1]} (x' - y_k)^2 - v_k. \tag{18}$$

Numerical results. We now study the numerical behavior of the stochastic optimization algorithm applied on the following one-dimensional example. The source distribution μ is the normalized Gaussian distribution of density $\rho(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, and the target distribution ν is the discrete uniform distribution on J equally spaced points between -1 and 1. An illustration of this setting is given in Fig. 1. In the following, x^* and v^* respectively refer to the optimal partition and optimal power weights computed explicitly as explained above.

There are several ways of evaluating the convergence of this algorithm, see Fig. 2. One quite simple way is to monitor along the iterations k the evolution of the relative ℓ^2-error

$$E_2(k) = \frac{\|v^k - v^* - \bar{v}^*\|}{\|v^* - \bar{v}^*\|} \tag{19}$$

where we remove the mean value $\bar{v}^*$ of v^* in order to cope with the invariance to constants (notice that by construction we have $\bar{v}^k = 0$). In this simple one-dimensional case we observed that E_2 decreases to zero. But let us recall that, in the non-strongly convex case, the convergence result for ASGD gives the convergence of the cost function $H(v^k)$ along the iterates, and not directly the convergence of v^k. For that reason, we also monitor in Fig. 2 the sequence $(H(v^k))$.

Another relevant way to monitor the convergence is to observe the distance between the transported measure $T_{v^k \sharp} \mu$ and the target measure ν. Since both measures

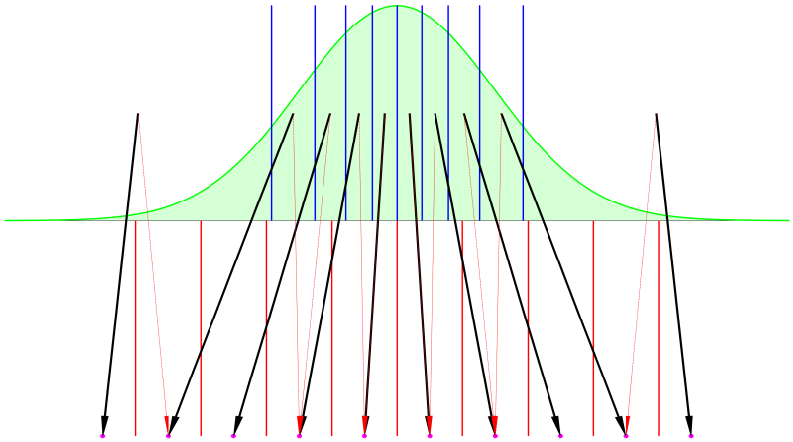

FIG. 1. ***1D semi-discrete OT illustration.*** *The green area represents the source Gaussian distribution μ, and the magenta dots represents the target distribution ν on the points (y_j). The vertical blue lines indicate the boundaries x_i^* of the optimal power cells $Pow_{v^*}(y_j)$, and the vertical red lines indicate the boundaries of the Voronoi cells $Pow_0(y_j)$. The corresponding optimal assignment T_v is displayed with black arrows, and the nearest-neighbor assignment T_0 is indicated with red dotted lines. One can observe that the OT assignment is very different from the NN projection. In particular, one may notice that a point y_j may not belong to the corresponding power cell.*

are discrete with same support S, we can easily compute the distance in total variation (TV) by

$$E_{\mathrm{TV}}(k) = \frac{1}{2}\sum_{y\in S} |T_{v^k\sharp}\mu(y) - \nu_y| = \frac{1}{2}\sum_{y\in S} |\mu(\mathrm{Pow}_{v^k}(y)) - \nu_y|. \tag{20}$$

This error is related to our optimization problem because $2E_{\mathrm{TV}}$ is exactly the ℓ^1-norm of the gradient of H (see (5)) and represents the amount of mistransported mass. But this TV error does not reflect how far the points have been mistransported. To account for the displacement error, one can of course rely on optimal transport distances. In Fig. 2 we monitor the L^1-Wasserstein distance E_{W_1} between $T_{v^k\sharp}\mu$ and ν which can be computed as L^1-distance between the cumulative distribution functions. The L^2-Wasserstein distance between $T_{v^k\sharp}\mu$ and ν was left aside here because it would require to solve a linear programming problem (which is quite long for $J = 1000$ points). However, recall that the optimal value $\max(H)$ is exactly the L^2 optimal transport cost between μ and ν. In other words, the cost function

$$H(v) = \sum_{y\in S}\int_{\mathrm{Pow}_v(y)} \|x-y\|^2 d\mu(x) + \sum_{y\in S} v(y)\big(\nu_y - \mu(\mathrm{Pow}_v(y))\big) \tag{21}$$

actually reflects the cost of the transport map T_v plus a term related to the constraint $\nu = T_\sharp\mu$ (which is asymptotically satisfied).

The results are displayed in Fig. 2. These graphs confirm that this stochastic optimization procedure is quite slow even in simple transportation cases. Still, one can obtain a relative ℓ^2-error around 10^{-3} in 10^8 iterations, even with 1000 points in the target distribution. When looking at the mass transportation problem, one can see that the results look very different depending on the adopted criterion. The TV distance (amount of mistransported mass) is of course the harder criterion and converges very slowly when J increases. However, in terms of H values or W^1 transport

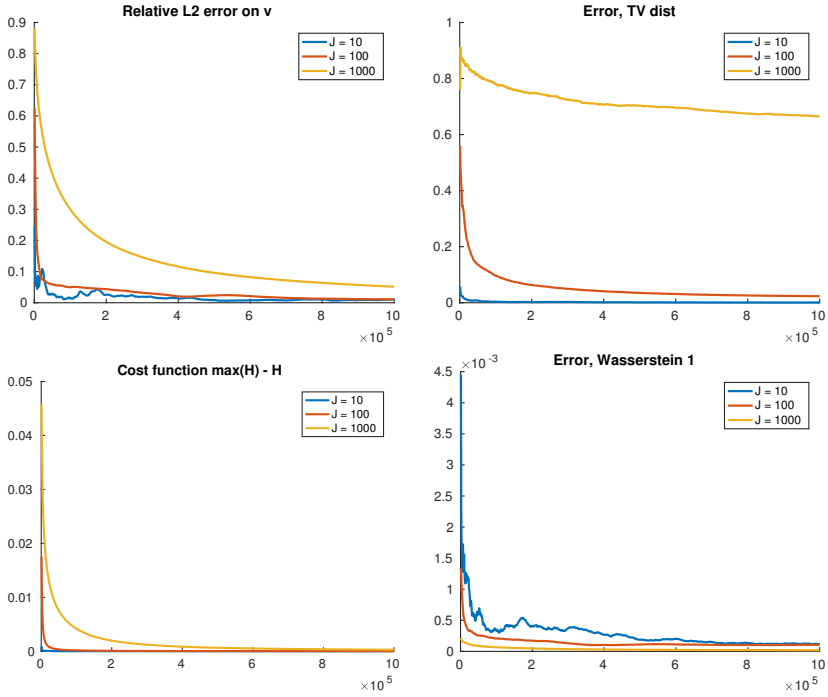

FIG. 2. ***Convergence of ASGD*** (8) ***in 1D depending on the number*** J ***of points in the target distribution.*** *Along the iterations* k*, we monitor the relative* ℓ^2*-error* E_2 *(top left), the values of the cost function* H *(bottom left), the TV-distance* E_{TV} *(top right), and the* L^1*-Wasserstein distance* E_{W1} *(bottom right). As expected, the convergence of ASGD for semi-discrete optimal transport becomes very slow when the number of points grows. Also, one may notice that the TV error goes to zero much more slowly than the other error measures, because it is oblivious of the points positions.*

cost, the convergence is relatively good, which means that, even if the convergence is slow on v, with many iterations we obtain a mapping T_v which is a reasonable approximation of the optimal transportation. It is also surprising to observe that for the W^1 transport cost, the convergence is actually faster for large J in this simple one-dimensional case.

2.3.2. D**-dimensional case study.** In higher dimension D, it is not possible anymore to draw explicit computations based on cumulative distribution functions. However, for very particular measures μ and ν, it is still possible to obtain a closed-form formula for an optimal set of weights v. Indeed, let us consider the semi-discrete optimal transport problem between

- μ, the uniform distribution on $[0,1]^D$.
- ν, the discrete uniform distribution on the $J = N^D$ points of the set

$$S = s\,.\left(t + \frac{1}{N-1}\{0,\dots,N-1\}^D\right) \subset \mathbb{R}^D$$

where $t \in \mathbb{R}^D$ is an offset vector, $s > 0$ is a scaling factor.

An illustration (for $D = 2$) of this optimal transport problem is given in Fig. 3.

One can write points in S using the index $i \in \{0,\dots,N-1\}^D$

$$y_i = s(\tfrac{i}{N-1} + t)\ .$$

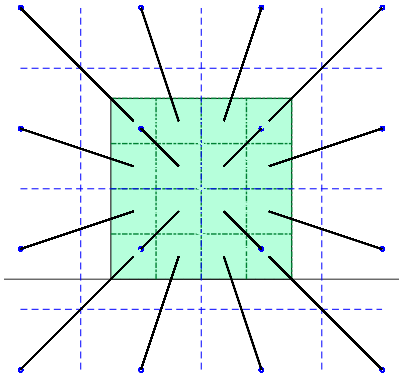

FIG. 3. ***D-dimensional semi-discrete OT illustration (here with*** $D = 2$***).*** *The source distribution* μ *is the uniform distribution on the green square, and the target discrete distribution* ν *is the uniform distribution on the blue dots. Green segments indicate the locations of the boundaries of the optimal power cells* Pow_{v^*}*, and the blue lines indicate boundaries of the Voronoi cells* Pow_0*. The correspond optimal assignment* T_{v^*} *is displayed with black lines.*

On the illustration, one sees immediately that, whatever the values of s and t, the optimal assignment consists in equally splitting the uniform mass of $[0,1]^D$ between the N^d hypercubes

$$\frac{i}{N} + [0, \tfrac{1}{N}]^D \quad (i \in \{0, \dots, N-1\}^d).$$

Then we can use the same methodology than in 1D to compute the optimal weights: for two adjacent points y_i, y_j there exists k such that $j = i + e_k$ and $y_j = y_i + \frac{s}{N-1} e_k$, and then the point $x_{ij} = \frac{j}{N}$ is at the boundary between the two corresponding power cells, and thus satisfies

$$v_j^* = v_i^* + \|y_j\|^2 - \|y_i\|^2 + 2\langle x_{ij}, y_i - y_j \rangle.$$

This allows to recursively compute the optimal weights v_i^*.

Here we will concentrate on the relative ℓ^2-error on v and illustrate the effect of the dimension by varying D from 1 to 6 while keeping approximately $J = N^D \approx 1300$ points in the target distribution (except for $D = 5$ and 6 for which $J = 1024, 729$ respectively). As one can observe on Fig. 4, the convergence gets slower and slower when the dimension increases. But still, we can get to 10^{-3} relative precision on v after 10^8 iterations, even in dimension 6. Besides, the differences between dimensions 4 to 6 seem to indicate that the convergence speed depends more on J (number of points in the target distribution) than on the dimension D.

In conclusion, even if it converges quite slowly, the stochastic gradient method for semi-discrete OT provides a reasonable approximation of the OT map even for high dimensions ($\gg 1$). In the following, we will see how to use optimal transportation in the patch space to enrich the Gaussian texture model.

3. Texture Synthesis Using Locally Transformed Gaussian Random Fields. In this section, we introduce a stochastic model for texture synthesis which consists in a local transform of a Gaussian random field. At the first level, the Gaussian random field captures long range correlations of the texture (but not its structured geometric features). Next, geometric structures are reimposed using a patch-based transformation. This patch-based transform is designed (offline) as an approximate

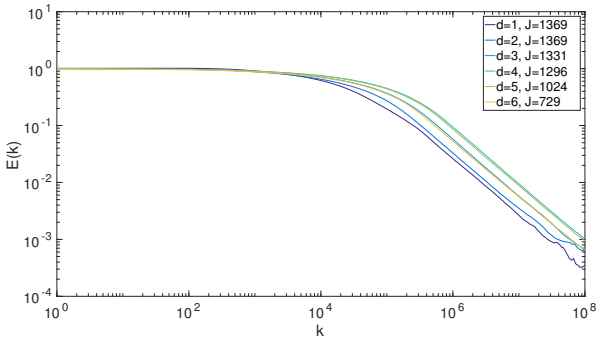

FIG. 4. ***Convergence of ASGD*** (8) ***depending on dimension*** D. *Here we use ASGD to solve the D-dimensional OT problem illustrated in Fig. 3. Along the iterations k, we monitor the evolution of the relative ℓ^2-error E_2. The curves are shown for dimensions $D = 1$ to 6, using approximately the same number of points J in the target distribution (indicated in the legend). The convergence gets slower when the dimension increases. But one can notice a slight gain when going from $D = 4$ to 5 or 6, which can be explained by the fact that we had to take a smaller J for $D = 5$ or 6.*

solution to a semi-discrete optimal transport problem, which ensures statistical compliancy with respect to the example. We first describe the monoscale version of the model which can be seen as an economical enrichment of the Gaussian model. We later propose a multiscale extension which allows to synthesize structured textures in a very efficient way while keeping statistical guarantees.

In this section, $u : \Omega \to \mathbb{R}^d$ is the exemplar texture defined on a domain $\Omega \subset \mathbb{Z}^2$.

3.1. Local transform of a Gaussian random field. As first level of synthesis, we propose to use the asymptotic discrete spot noise (ADSN) [14, 15] defined as

$$\forall\, x \in \mathbb{Z}^2,\ U(x) = \bar{u} + \sum_{y \in \mathbb{Z}^2} t_u(y) W(x-y) \quad \text{where} \quad \begin{cases} \bar{u} = \frac{1}{|\Omega|} \sum u(x), \\ t_u = \frac{1}{\sqrt{|\Omega|}} (u - \bar{u}) \mathbf{1}_\Omega \end{cases} \tag{22}$$

and where W is a normalized Gaussian white noise on $\mathbb{Z}^2$. This random field U is a stationary Gaussian random field whose first and second order moments are the empirical mean and covariance of the exemplar texture. Thus, U can be considered as a "Gaussianized" version of u, which have the correct correlations but no salient structures.

Next we propose to apply a local transform $T : \mathbb{R}^D \to \mathbb{R}^D$ which operates in the patch space $\mathbb{R}^D$ where $\omega = \{0, \dots, w-1\}^2$ is the patch domain ($w \in \mathbb{N}^*$), and $D = dw^2$. This mapping T is applied to each patch of the Gaussian synthesis U, and an image V is later recomposed by averaging: the value at pixel x is the average of values of x in all overlapping patches. In other words, for fixed non-negative weights $(\theta(h))_{h \in \omega}$ whose sum equals 1, we define the transformed random field V as

$$\forall x \in \mathbb{Z}^2, \quad V(x) = \sum_{h \in \omega} \theta(h) T(U_{|x-h+\omega})(h). \tag{23}$$

In practice, we generally use $\theta = \frac{1}{|\omega|} \mathbf{1}_\omega$ (simple average). In the following we state some properties of such a transformed random field.

PROPOSITION 3. *V is a stationary random field on $\mathbb{Z}^2$ and satisfies the following long-range independence property: if S denotes the finite support of the auto-correlation function*

$$a_u(z) = \sum_{x\in\mathbb{Z}^2} t_u(z)t_u(x+z)^T \tag{24}$$

then for every $A, B \subset \mathbb{Z}^2$ such that $(A-B)\cap(S+4\omega)=\emptyset$ the restrictions $V_{|A}, V_{|B}$ are independent.

Proof. The covariance of U is $Cov(U)(x-y) = \mathbb{E}[(U(x)-\bar{u})(U(y)-\bar{u})] = a_u(y-x)$. Since U is Gaussian, we get that $U(x) \perp\!\!\!\perp U(y)$ as soon as $x-y \notin S$. Therefore, if $x-y \notin S+2\omega$, then $U_{|x+\omega} \perp\!\!\!\perp U_{|y+\omega}$ and thus $P_x \perp\!\!\!\perp P_y$. After averaging we get $V(x) \perp\!\!\!\perp V(y)$ as soon as $x-y \notin S+4\omega$. The generalization to subsets A, B is straightforward. ∎

This property is a guarantee of spatial stability for synthesis, meaning that the corresponding synthesis algorithm will not start to "grow garbage" as may do the method of [13]. Next we give another property which allows to control the difference between U and V in terms of medium-range correlations. For that we need the following lemma.

LEMMA 4. *Let F, G be two real-valued stationary random fields defined over $\mathbb{Z}^2$ with respective standard deviations σ_F and σ_G. Then, for all $t \in \mathbb{Z}^2$,*

$$|\operatorname{Cov}(G)(t) - \operatorname{Cov}(F)(t)| \le (\sigma_F + \sigma_G)\operatorname{Var}(G(0)-F(0))^{\frac{1}{2}}.$$

Proof. This is an elementary proof that solely uses the bilinearity of the covariance and the Cauchy-Schwarz inequality.

$$\begin{aligned}\operatorname{Cov}(G)(t) &= \operatorname{Cov}(G(t),G(0)) = \operatorname{Cov}(G(t)-F(t),G(0)) + \operatorname{Cov}(F(t),G(0))\\ &= \operatorname{Cov}(G(t)-F(t),G(0)) + \operatorname{Cov}(F(t),G(0)-F(0)) + \operatorname{Cov}(F)(t)\end{aligned}$$

Hence,

$$\begin{aligned}&|\operatorname{Cov}(G)(t) - \operatorname{Cov}(F)(t)| = |\operatorname{Cov}(G(t)-F(t),G(0)) + \operatorname{Cov}(F(t),G(0)-F(0))|\\ &\le |\operatorname{Cov}(G(t)-F(t),G(0))| + |\operatorname{Cov}(F(t),G(0)-F(0))|\\ &\le \operatorname{Var}(G(t)-F(t))^{\frac{1}{2}}\operatorname{Var}(G(0))^{\frac{1}{2}} + \operatorname{Var}(F(t))^{\frac{1}{2}}\operatorname{Var}(G(0)-F(0))^{\frac{1}{2}}\\ &= (\sigma_F+\sigma_G)\operatorname{Var}(G(0)-F(0))^{\frac{1}{2}}.\end{aligned}$$ ∎

Remark: In the above lemma, $F(0)$ can be replaced by $F(h)$ for $h \in \mathbb{Z}^2$ by changing F in $F(\cdot+h)$.

We can now apply this lemma to the random fields U and V used in our model.

PROPOSITION 5. *Recall that U is a stationary Gaussian field, and that V is a local transform of U using the patch operator $T : \mathbb{R}^D \to \mathbb{R}^D$ (see (23)). We assume that $d = 1$ (i.e. U and V are real-valued). Then, for all $x \in \mathbb{Z}^2$,*

$$\mathbb{E}((V(x)-U(x))^2) \le \|\theta\|_\infty \mathbb{E}\left(\|T(U_{|\omega}) - U_{|\omega}\|^2\right). \tag{25}$$

Consequently, denoting σ_U and σ_V the standard deviations of U and V respectively,

$$\forall t \in \mathbb{Z}^2, \quad |\operatorname{Cov}(V)(t) - \operatorname{Cov}(U)(t)| \le (\sigma_U+\sigma_V)\sqrt{\|\theta\|_\infty}\mathbb{E}\left(\|T(U_{|\omega}) - U_{|\omega}\|^2\right)^{\frac{1}{2}}. \tag{26}$$

Proof. First, by convexity of $s \mapsto s^2$, for all $x \in \mathbb{Z}^2$,

$$\begin{aligned}(V(x) - U(x))^2 &= \left(\sum_{h\in\omega} \theta(h)(T(U_{|x-h+\omega})(h) - U(x))\right)^2 \\ &\leq \sum_{h\in\omega} \theta(h)(T(U_{|x-h+\omega})(h) - U(x))^2.\end{aligned}$$

Now, for all $h \in \omega$, the difference $T(U_{|x-h+\omega})(h) - U(x)$ is the h-coordinate of the patch difference $T(U_{|x-h+\omega}) - U_{|x-h+\omega}$. More formally,

$$T(U_{|x-h+\omega})(h) - U(x) = T(U_{|x-h+\omega})(h) - U_{|x-h+\omega}(h) = (T(U_{|x-h+\omega}) - U_{|x-h+\omega})(h).$$

For N an integer larger than 1 let us denote by A_N the discrete square $A_N = \{0, \dots, N-1\}^2$ and note that $|A_N| = N^2$. Thus we have

$$\begin{aligned}\sum_{x\in A_N} (V(x) - U(x))^2 &\leq \sum_{h\in\omega} \sum_{x\in A_N} \theta(h)((T(U_{|x-h+\omega}) - U_{|x-h+\omega})(h))^2 \\ &\leq \sum_{h\in\omega} \sum_{y\in -h+A_N} \theta(h)((T(U_{|y+\omega}) - U_{|y+\omega})(h))^2.\end{aligned}$$

Now remark that for all $h \in \omega$ we have the inclusion $-h + A_N \subset A_N \oplus (-\omega)$. Hence

$$\begin{aligned}\sum_{x\in A_N} (V(x) - U(x))^2 &\leq \sum_{h\in\omega} \sum_{y\in A_N\oplus(-\omega)} \theta(h)((T(U_{|y+\omega}) - U_{|y+\omega})(h))^2 \\ &\leq \sum_{y\in A_N\oplus(-\omega)} \sum_{h\in\omega} \theta(h)((T(U_{|y+\omega}) - U_{|y+\omega})(h))^2 \\ &\leq \sum_{y\in A_N\oplus(-\omega)} \|T(U_{|y+\omega}) - U_{|y+\omega}\|_\theta^2,\end{aligned}$$

introducing the notation

$$\|P\|_\theta^2 = \sum_{h\in\omega} \theta(h)P(h)^2$$

for a patch $P \in \mathbb{R}^\omega$. Using the stationarity of U and V we get

$$\begin{aligned}\mathbb{E}((V(0) - U(0))^2) &= \frac{1}{|A_N|} \sum_{x\in A_N} \mathbb{E}((V(x) - U(x))^2) \\ &\leq \frac{1}{|A_N|} \sum_{y\in A_N\oplus(-\omega)} \mathbb{E}(\|T(U_{|y+\omega}) - U_{|y+\omega}\|_\theta^2) \\ &\leq \frac{|A_N \oplus (-\omega)|}{|A_N|} \mathbb{E}(\|T(U_{|\omega}) - U_{|\omega}\|_\theta^2).\end{aligned}$$

Letting N tends to $+\infty$, we obtain that

$$\mathbb{E}((V(0) - U(0))^2) \leq \mathbb{E}(\|T(U_{|\omega}) - U_{|\omega}\|_\theta^2).$$

Now note that $\|P\|_\theta^2 \leq \|\theta\|_\infty \|P\|^2$ to obtain the enunciated result. ∎

3.2. Optimal transport in patch space. Now that we have fixed a generic framework of locally transformed random fields, an important point is to wisely choose the patch transform T in order to reimpose the statistics of the exemplar texture on local features. For that we choose an OT map between the distribution of the Gaussian patches of U and the empirical patch distribution of the exemplar texture. The need of optimality in this transformation can be understood in this way: we want to change the patches to get the proper patch distribution, but with the least possible changes in order to keep the (second-order) statistical control obtained in the Gaussian field. This heuristics reflects in the fact that the optimal transport map T actually minimizes the right-hand side of (26). In order to stay in a reasonable framework for stochastic optimal transport, we will only work with 3×3 patches.

The adopted point of view is to consider that all statistics on local features are encoded in the patch distribution at multiple scales. In addition to the color distribution, the 3×3 patch distribution encompasses the joint distributions of all 3×3 differential filters, e.g. the distributions of x or y derivatives, the distribution of the Laplacian, the density of oriented edges, the correlations between those derivatives, and so on. In this paragraph we explain the monoscale model that uses one OT map to reimpose the 3×3 patch distribution, and we will explain the multiscale extension in Section 3.3.

More precisely, the source distribution here is the distribution μ of $U_{|\omega}$, that is the distribution of any Gaussian patch of U (thanks to stationarity). Since the covariance function of U is given by (24), we can explicitly compute the parameters of μ. Notice that except in degenerate cases (that rarely happen in practice), μ is absolutely continuous with respect to the Lebesgue measure.

On the other hand, an ideal target measure is the empirical distribution of patches of the exemplar texture u, that is,

$$\nu_{\text{emp}} = \frac{1}{|P(u)|} \sum_{p \in P(u)} \delta_p, \quad \text{where} \quad P(u) = \{\, u_{|x+\omega} \mid x + \omega \subset \Omega \,\}. \tag{27}$$

Unfortunately, texture images generally contain much more than 10 000 patches which is not a reasonable framework for ASGD-based OT in terms of computational time. Thus, we propose to approximate the empirical measure with the subsampled distribution

$$\nu = \frac{1}{J} \sum_{j=1}^{J} \delta_{p_j}, \tag{28}$$

where $p_1, \dots, p_J$ are $J = 1000$ patches which are uniformly drawn from $P(u)$. Of course, if $|P(u)| < 1000$, we do not need this subsampling step and take $\nu = \nu_{\text{emp}}$. As will be discussed in Section 4.3, this subsampling step, although quite naive, is in practice sufficient to account for the variability of the target measure, and compares well to more involved procedures based on clustering.

Thus, μ and ν are two probability measures on $\mathbb{R}^D$ with $D = dw^2$. Using the algorithm explained in Section 2.2, we compute the optimal assignment T that realizes the semi-discrete OT from μ to ν. Let us recall that this optimal assignment $T = T_v$ is a weighted NN assignment defined by

$$T_v(p) = \operatorname*{argmin}_{(p_j)_{j=1,\dots,J}} \|p - p_j\|^2 - v_j \tag{29}$$

where v is an optimal set of weights computed with ASGD. Each iteration of ASGD needs one sample of the source distribution μ, which amounts here to sample a patch of the Gaussian random field U. Of course, instead of computing the entire synthesis with DFT and extract a patch, it is much more efficient to explicitly store the covariance matrix of $U_{|\omega}$ and to use it for sampling[2]. Let us mention that for 3×3 color patches, the patch space has dimension 27.

This monoscale texture model is illustrated in Fig. 5. On this figure, one can observe the effect of reimposing the patch distribution through the local transform T. This monoscale model is an interesting enrichment of the Gaussian model U, if only because it precisely respects the marginal color distribution of the original image. Of course, in comparison to the NN projection, the benefit of the statistical control is obvious (which will be confirmed in Section 4.3). The OT model also clearly better respects the density of oriented edges (but it is not trivial for a human observer to precisely evaluate this fact in a manner that is clearly independent of its sensitivity to the color distribution).

One can also observe that some mid-range correlations persist after the local transformation T. This empirically confirms the result of Proposition 5. But the inequality obtained in this proposition is actually too loose to provide a fine control of the mid-range correlations. Indeed, for several textures, we evaluated the quality of the inequality (26) with Monte-Carlo simulations. We found that for small shifts $t \in \mathbb{Z}^2$, the bound provided by the right-hand side of (26) has the same order of magnitude as the input covariance $\mathrm{Cov}(U)(t)$ (and is twice smaller in the best cases). But the bound gets very bad for large shifts t, which could be expected because for many textures $\mathrm{Cov}(U)(t)$ decreases quickly when $|t|$ increases (whereas the right-hand side of (26) does not depend on t). In any case, the empirical values obtained for the left-hand side are much smaller than the generic bound of this proposition.

3.3. Multiscale extension. Now, we propose a multiscale extension of the previous model. Starting from a Gaussian synthesis at a coarse scale, we recursively apply local transforms in order to reimpose the patch distributions at different scales. In order to go from one scale to the next one, we use a simple upsampling procedure which consists in pasting the same patch taken at two adjacent scales.

Notation. In order to adapt the previous algorithm in a multiscale fashion, we need to compute as input

- subsampled versions u^ℓ of the original image u at scales $\ell = 0, \dots, L-1$,
- the empirical patch distribution ν^ℓ at each scale (approximated again by randomly picking 1000 patches in u^ℓ)

The images u^ℓ are obtained by successive subsampling of $u = u^0$ by a factor of 2 (obtained by bicubic averaging). We use the convention that u^ℓ is defined on the subgrid

$$\Omega^\ell = \Omega \cap 2^\ell \mathbb{Z}^2,$$

so that a coordinate $y \in \Omega^\ell$ also appears in the adjacent finer grid with the same notation $y \in \Omega^{\ell-1}$.

Initialization at the coarsest scale $\ell = L-1$. At the coarsest scale $\ell = L-1$, we use the monoscale model explained in Section 3.2. We compute the ADSN model U^ℓ on $2^\ell\mathbb{Z}^2$ associated to u^ℓ, and the corresponding Gaussian patch distribution μ^ℓ. And then, as in (23), we apply a local transform T^ℓ which realizes the OT from μ^ℓ to ν^ℓ,

[2]Since we need many samples, we actually store the Cholesky decomposition of the covariance, which is all is need for Gaussian sampling (besides the mean value).

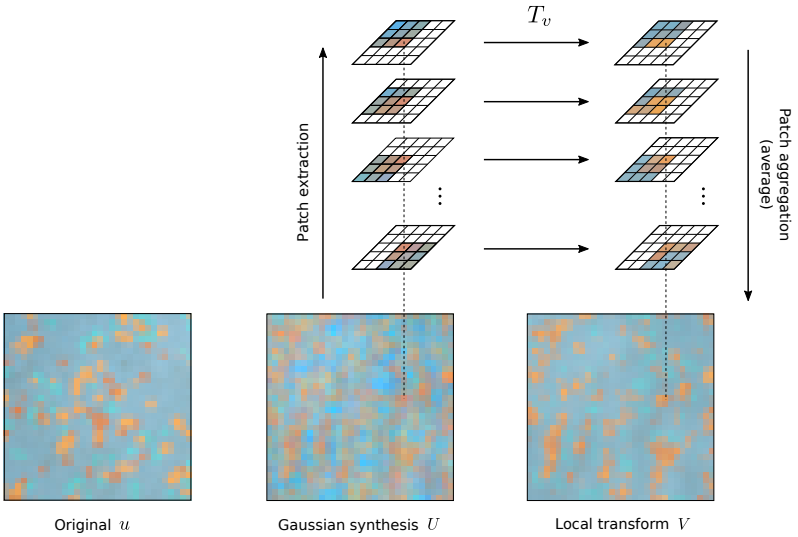

FIG. 5. ***Texture synthesis with a local transform of a Gaussian field.*** *In this figure we illustrate the use of the monoscale model for synthesis. We display the original texture (left), the synthesis U at the first level using the ADSN model, and the random field V obtained after local transformation of U using an optimal assignment T_v in the patch space.*

so that we get a random field

$$V^\ell(x) = \sum_{h\in 2^\ell\omega} \theta(\tfrac{h}{2^\ell}) T^\ell(U^\ell_{|x-h+2^\ell\omega})(h), \quad x \in 2^\ell\mathbb{Z}^2. \tag{30}$$

Again this OT map is actually a patch assignment: at the position x in the synthesis, we will use the patch taken in u^ℓ at position $Y^\ell(x)$ (see Fig. 6 for an illustration). We thus get a "coordinate map" $Y^\ell : 2^\ell\mathbb{Z}^2 \to \Omega^\ell$ which allows to write T^ℓ and V^ℓ as

$$T^\ell(U^\ell_{|x+2^\ell\omega}) = u^\ell_{|Y^\ell(x)+2^\ell\omega}, \quad x \in 2^\ell\mathbb{Z}^2, \tag{31}$$

$$V^\ell(x) = \sum_{h\in 2^\ell\omega} \theta(\tfrac{h}{2^\ell}) u^\ell\big(Y^\ell(x-h)+h\big), \quad x \in 2^\ell\mathbb{Z}^2. \tag{32}$$

Then we upsample the current synthesis using the twice larger patches at the same positions in the next scale

$$\forall x \in 2^\ell\mathbb{Z}^2,\ \forall s \in \{0, 2^{\ell-1}\}^2, \quad U^{\ell-1}(x+s) = \sum_{h\in 2^\ell\omega} \theta(\tfrac{h}{2^\ell}) u^{\ell-1}\big(Y^\ell(x-h)+h+s\big). \tag{33}$$

Iterating local transforms at the next scales. Suppose that the model U^ℓ has been computed at scale $\ell \in \{1, \dots, L-2\}$. Then we estimate a Gaussian mixture model (GMM) μ^ℓ with n_{GMM} components that fits the patch distribution of U^ℓ. For that we use standard implementations of the expectation-maximization algorithm adapted to the GMM case [42]. Using the estimated μ^ℓ and the corresponding sampling function[3], we compute an OT map T^ℓ from μ^ℓ to ν^ℓ with ASGD, as in the monoscale case. Once T^ℓ has been computed, the transformed random field V^ℓ and its upsampled version $U^{\ell-1}$ are obtained with the same formulae (30), (31), (32), (33).

[3]Here again, it is of course more efficient to compute the Cholesky decompositions of the Gaussian components covariances once and for all.

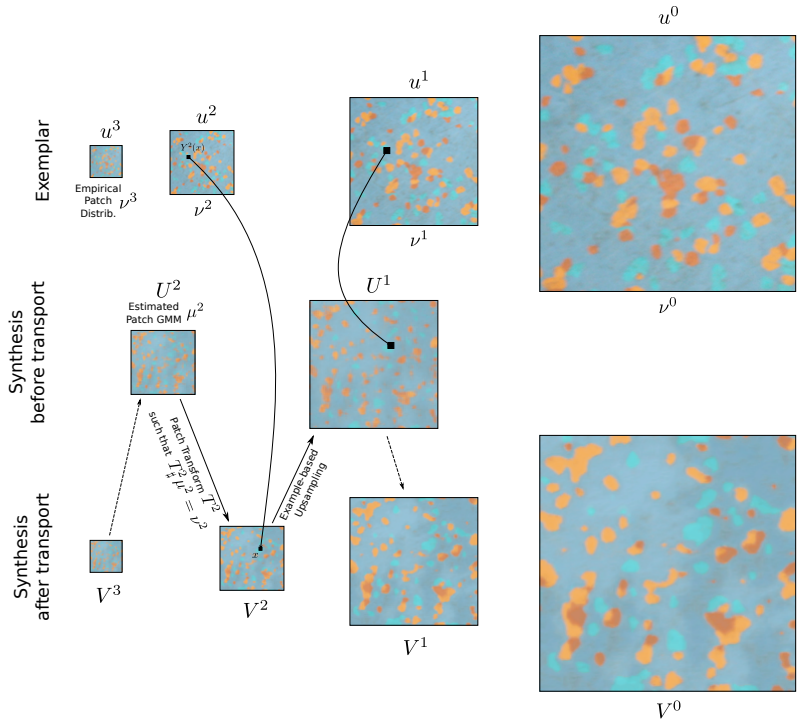

FIG. 6. ***Multiscale synthesis.*** *In this figure we illustrate the proposed multiscale synthesis method, applied here with* 4 *scales. In the first row we display the original images* u^ℓ *at each scale. In the last row we display the synthesized images* V^ℓ *at each scale. In the middle row, we display the synthesized images* U^2 *and* U^1 *(obtained before patch OT) in order to illustrate how to go from scale* 2 *to scale* 1*. Notice in particular that the patch transform essentially assembles patches taken from the exemplar (by averaging) and that the exemplar-based upsampling step takes the patches at the same position, but twice larger.*

Remarks. The OT maps T^ℓ are computed once and for all during the "model estimation", which consists in one first synthesis pass (setting the output size as the exemplar for instance). In other words, after the estimation, for all scales ℓ we store T^ℓ, that is, the corresponding 1000 patches p^ℓ_j (of size 3×3) of the exemplar u^ℓ, the associated weights v and the upsampled patches P^ℓ_j (of size 6×6). Once the model has been estimated, all these maps T^ℓ can be evaluated on the fly.

In conclusion to this section, let us emphasize that imposing the patch distribution at each scale with OT can thus be thought of as a very non-parametric way of imposing wavelet statistics. However, the correlations between adjacent scales are not directly addressed with these local transforms, but are more or less preserved with the adopted example-based upsampling of (33).

4. Results and discussion. In this section, we provide several synthesis results obtained with the monoscale and multiscale models presented in the previous sections, which demonstrate that this model allows for fast synthesis of structured textures. We empirically confirm the benefit of applying well-designed local transforms to enforce the patch distribution (and study the impact of the average recomposition step). We compare with the simpler alternative which consists in iterating patch NN projections at each scale, and thus demonstrate that applying one OT map leads to visually better

results (thanks to the global statistical control), while being much faster. We discuss the two main parameters of the multiscale model, i.e. the number of scales L, and the number n_{GMM} of components used in the GMM. We also discuss the number of patches used in the target discrete patch distributions, and compare with other simple measure quantization technique. Using several exemplar textures, we compare this multiscale model with several other models and algorithms for texture synthesis.

4.1. Monoscale model. In Fig. 7, we display several synthesis results obtained with the monoscale model explained in Section 3.2. One can observe that applying the OT-based local transform is a way to enrich the Gaussian model which is statistically more relevant than the NN projection. Let us recall that the NN projection is here performed on the same subset of 1000 patches than OT (i.e. NN projection consists in taking $v = 0$ in Equation (1)). This monoscale model is interesting to synthesize slightly structured textures. In particular, for the second and third examples of Fig. 7, the input color distribution is not symmetric around the mean (because of shade effects), and the OT patch transform allows to break the symmetry of the Gaussian model in a better way than the NN projection.

In Fig. 7, one can also apprehend the importance of capturing the mid-range correlations with an adapted Gaussian field as input to the local transform. If we use a trivial Gaussian white noise instead, it is still possible to compute a relevant OT map (that realizes the semi-discrete OT between the Gaussian white noise patch μ and the exemplar patch distribution ν), but then the transformed random field only looks like a very slightly structured noise. Indeed, pixels at distance $> 4\sqrt{2}$ are still independent.

In Fig. 8, we empirically confirm that the OT-based local transform allows to reimpose the patch statistics. For 3×3 color patches, the patch space has dimension 27 so an appropriate way to visualize the patch distributions is to monitor the one-dimensional distributions obtained in the principal components of the exemplar patch distribution. Again, these diagrams confirm that the distribution of the transformed patches is approximately the same as the exemplar patch distribution. The two sources of approximation are the quantization of the target distribution (by randomly picking 1000 patches in the exemplar texture) and the fact that ASGD has not fully converged. Also, on Fig. 8, one can see that the average recomposition step does not drastically change the patch distribution. Heuristically (think for example in terms of marginal color distribution) this average step will tend to concentrate the patch distribution, but for 3×3 the amount of induced blur is reasonable (because each patch is merged with only 8 neighbors that may already be compatible). However, as will be observed later, this step induces a slight loss in the textural grain.

4.2. Multiscale model. Here, we present and comment several texture synthesis results obtained with the multiscale OT model. We display the successful results in Fig. 6 and Fig. 9 and the relative failures in Fig. 10. All these examples were generated with $L = 4$ scales and $n_{\mathrm{GMM}} = 4$ components in the GMM source distribution at each scale. The example of Fig. 6 clearly illustrates the effect of imposing the patch distribution at each scale. In Fig. 9, one can observe that this model is able to synthesize structured textures provided that the structural elements of the exemplar are sufficiently repeated in the input image. Of course, this model works particularly well with textures exhibiting structures similar to excursions of Gaussian fields (see e.g. [30]), like the one of Fig. 6 and the top right example of Fig. 9. Notice that all the exemplar textures of Fig. 9 possess geometric elements that can be slightly deformed without perturbing the perception. On any other such texture, the model is expected

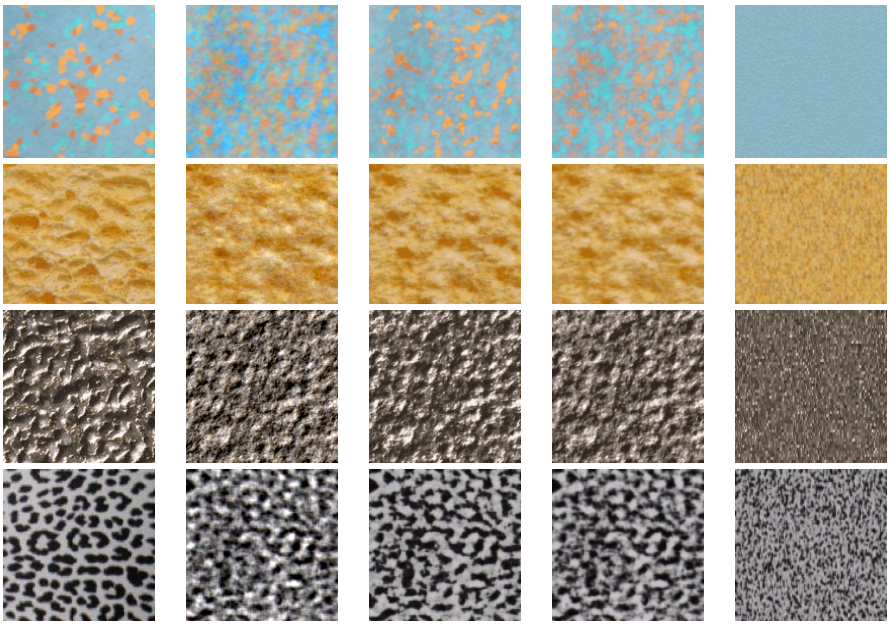

FIG. 7. ***Monoscale synthesis.*** *In this figure we display synthesis results obtained with locally transformed Gaussian random fields. For each row, column 1 contains an exemplar texture; column 2 contains samples of the associated ADSN model; column 3 and 4 contain local transformations of the ADSN obtained with the* 3×3 *patch optimal assignment* T_v *(OT) or patch nearest neighbor projection* T_0 *(NN); column 5 contains a local transformation of a Gaussian white noise (WN) with an optimal patch assignment adapted to the white noise input. The OT assignment better preserves patch statistics than the NN projection. Besides, the last column illustrates the importance to start from a spatially correlated Gaussian model at the first level.*

to behave perfectly.

In contrast, several textures of Fig. 10 have a very constrained local geometry which cannot be reimposed properly by working only with 3×3 patches. Such a property may cause synthesis defects detected after attentive examination (even if some of these examples can be seen as success with only pre-attentive examination). At first, one may attribute this limitation to the brutal subsampling of the exemplar distribution to 1000 patches (which would not be sufficient to reconstruct all the edges of the structural elements of the texture, especially if there is variability in color like in the examples in the last row of Fig. 10). But surprisingly, this may not be the main reason. Indeed, in Fig. 11, we show that using all patches of the exemplar does not help to recover the local geometry of the exemplar in a cleaner way. More generally, one may very well question the way to subsample the target patch distribution. We experimentally proposed a fixed number of 1000 patches for the discrete target 3×3 patch distributions. In our work, this choice is motivated by the convergence study of ASGD (see Section 2.3): we took the maximal order of magnitude for which the convergence of ASGD is clearly visible (in high dimension) with reasonable computational time. The relevance of this choice is confirmed by Fig. 11 which shows that taking more than 1000 patches does not increase the variability (and actually slightly decreases it because ASGD converges more slowly). Therefore, we emphasize that this value must not be understood as a parameter of the model. As a collateral benefit, all

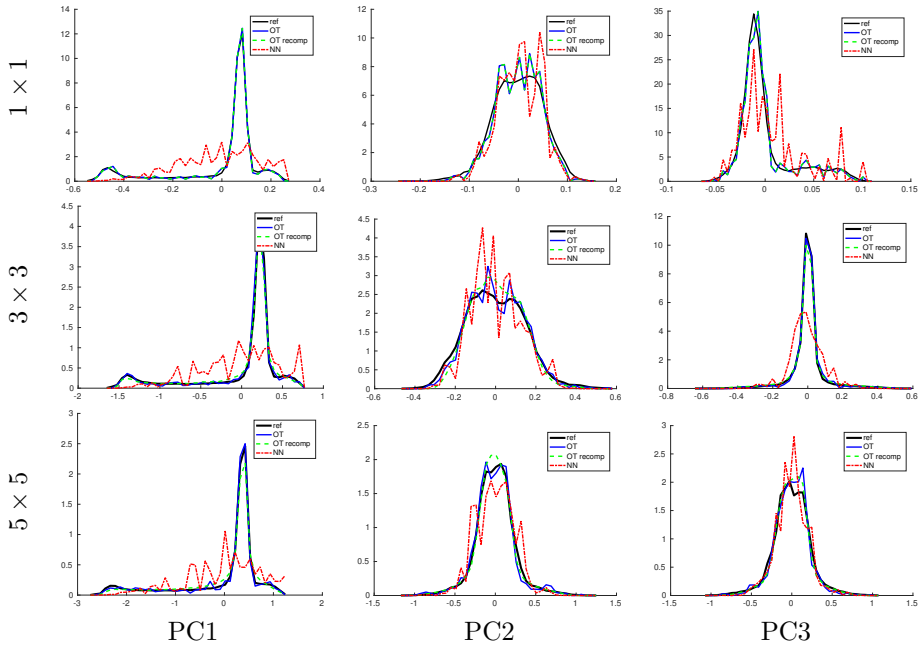

FIG. 8. ***Patch distribution (three first principal components).*** *For the first image of Fig. 6 we plot the estimated distribution of patches in the three first principal components (columns) for different patch sizes (rows). The PCA transform is obtained on the exemplar patch distribution. We compare the patch distributions of the exemplar image (ref), of the synthesized image before patch recomposition (OT) and after (OT recomp), and of the transformed patch with nearest-neighbor projection (NN). Even if we only approximate the OT mapping, it suffices to reproduce the reference patch distribution better than the NN projection.*

the synthesis experiments shown in this paper suggests that 1000 is a good order of magnitude for quantizing any 3×3 patch distribution extracted from a texture image. Of course this choice impacts the computation time for synthesis (because each local transform computes a weighted NN projection on 1000 patches).

On the other hand, with a budget of 1000 patches, one may imagine other techniques to quantize the target distribution. This can be seen as a clustering issue for which random subsampling provides a (quite naive and yet) reasonable solution, keeping in mind that the chosen patches are used for synthesis. For example, one may use a different target measure $\nu = \sum_{j=1}^{J} \nu_j \delta_{q_j}$ obtained by first clustering the exemplar patch distribution with a k-means algorithm with $J = 1000$ clusters, and then computing the nearest patch q_j of each centroid and the proportion ν_j of points in the cluster. We experimented this refined subsampling strategy, but unfortunately it does not improve the visual quality of the output as illustrated in Fig. 11.

Another possibility to overcome the loss of geometric structures is to work with larger patches, which would allow to copy larger pieces of the exemplar. But for now this is rather impractical for the following technical reasons: firstly, in very high dimension, ASGD converges too slowly which makes it impractical (see Fig. 4); secondly, the average recomposition step would introduce too much blur and should be replaced; finally, the target exemplar distribution would be much more complex and thus would require a larger subsample set of patches.

Again, this multiscale model can be seen as a rich extension of the Gaussian model (which is only adapted to microtextures [14]). Thus one natural condition for this

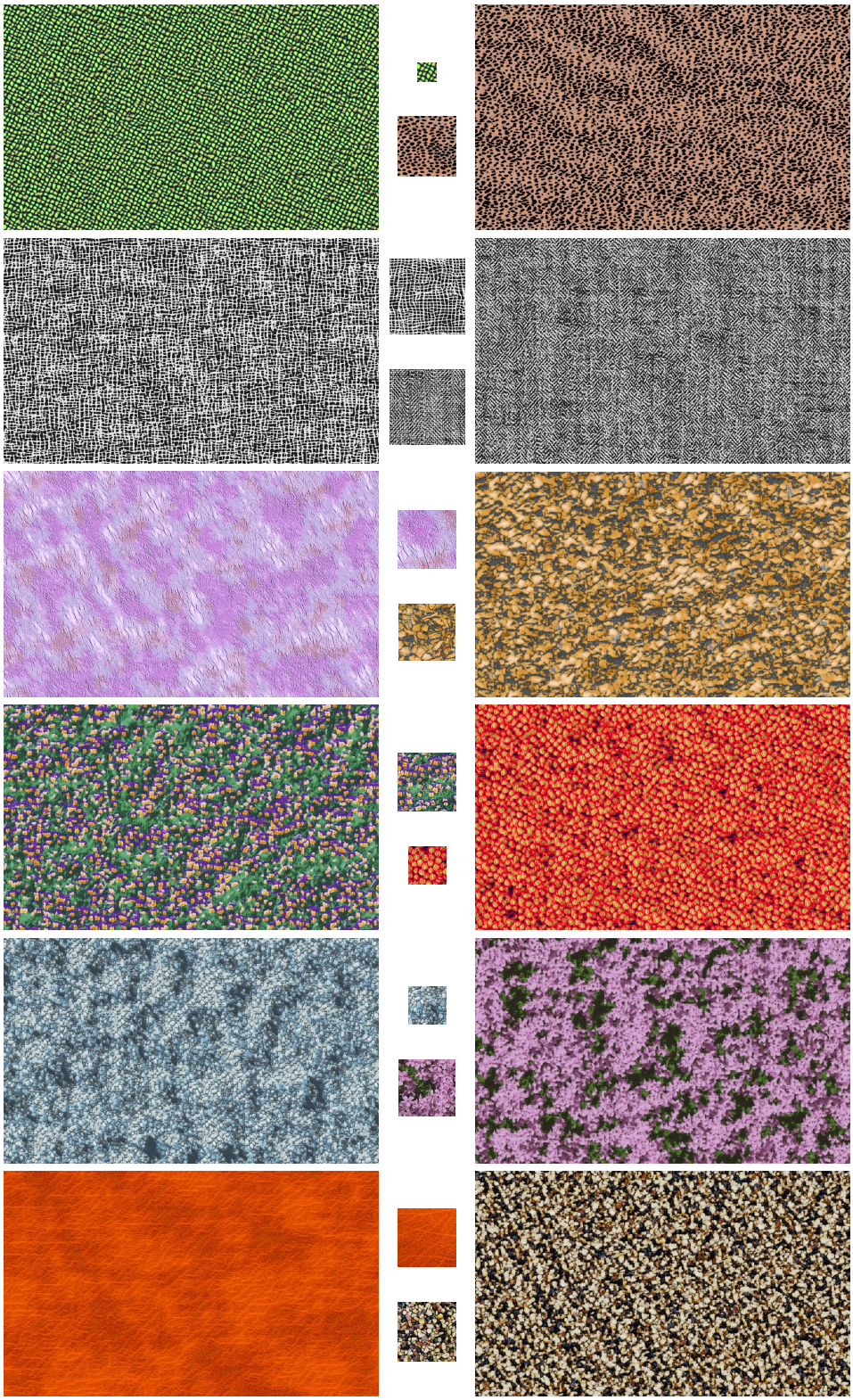

FIG. 9. ***Multiscale synthesis, successful cases.*** *For each exemplar texture shown in the middle column we display a synthesized texture of size* 1280×768*. We used* $L = 4$ *scales, and* $n_{GMM} = 4$ *Gaussian components at each scale. See the text for comments.*

FIG. 10. ***Multiscale synthesis, failure cases.*** *For each exemplar texture shown in the middle column we display a synthesized texture of size* 1280×768. *We used* $L = 4$ *scales, and* $n_{GMM} = 4$ *Gaussian components at each scale. See the text for comments.*

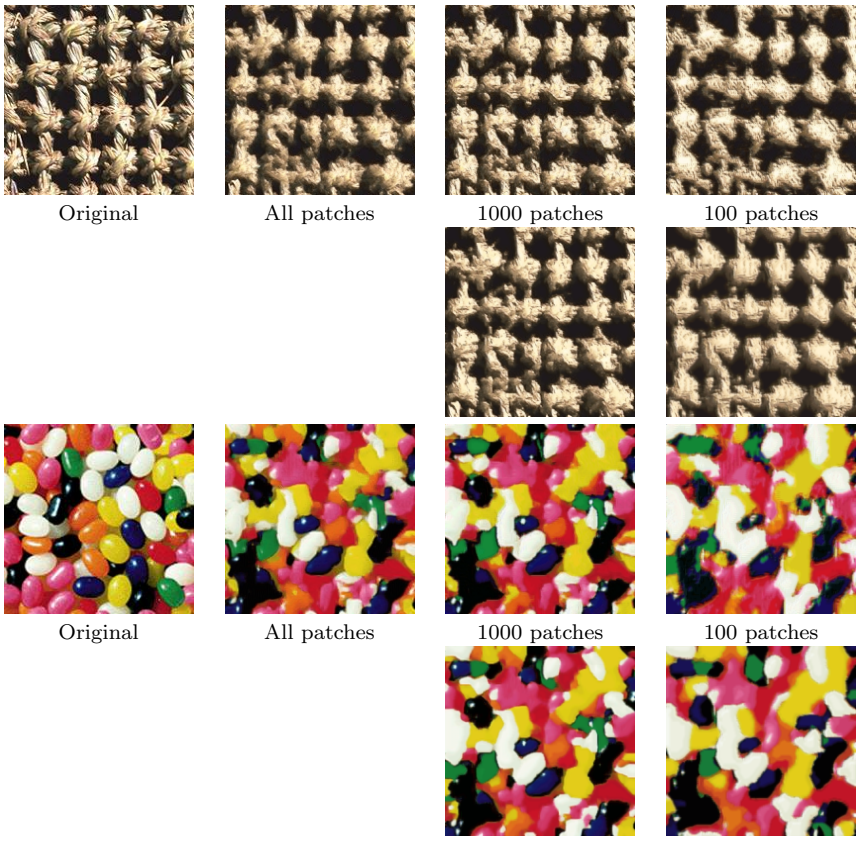

FIG. 11. ***Subsampling the target patch distributions.*** *In this experiment we question the target patch distribution ν. In the first row, we display results obtained with simple subsampling with all available patches (2nd column),* 1000 *patches (3rd column) and* 100 *patches (4th column). In the second row, we use other discrete distributions supported on NN of k-means centroids applied with* 1000 *patches (3rd column) and* 100 *patches (4th column). Let us mention that these images have* $\approx$ 30000 *patches at scale* 0, $\approx$ 9000 *patches at scale* 1, $\approx$ 2200 *patches at scale* 2 *and* $\approx$ 500 *patches at scale* 3*. See the text for comments.*

multiscale model to fit a given exemplar texture is that the exemplar at coarse scale u^{L-1} is a microtexture. This may explain some failures on examples where there is still too much structure at coarse scale, and thus defects of the coarse ADSN may transfer to the fine scale synthesis. For the quasi-periodic examples (Fig 10, fifth row), because of the frequential discretization involved in the discrete Fourier transform, interference patterns appear at the coarsest scale, which cannot be corrected by the further local transforms.

Let us also mention that we observe a slight loss in textural details on several textures (e.g. on Fig. 6 and on the "rope" example in the second row of Fig. 10). Since the OT map reimposes the marginal statistics, such a loss can only be attributed to the average recomposition step. One can attenuate this artifact by using a simple trick which consists in changing the patch recomposition strategy at scale 0 (that is, taking $\theta = \delta_0$ in formula (23)). With this simple modification, of course we better recover the marginal distribution, and thus the textural grain, as can be observed in Fig. 12. But this may also introduce other staircasing-like artifacts.

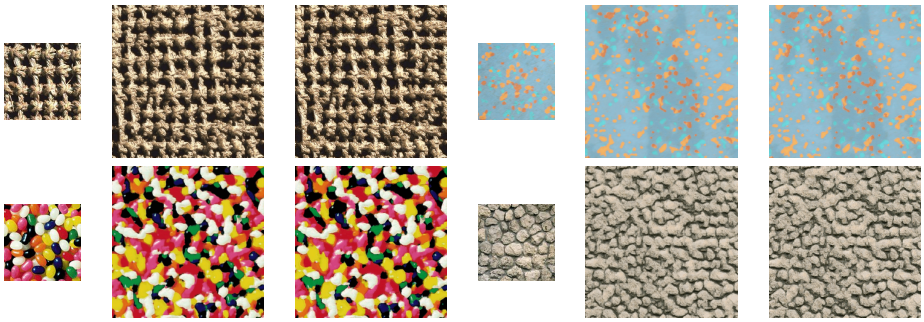

FIG. 12. ***Changing patch recomposition strategy at scale*** 0**.** *In this figure we illustrate that the slight textural loss observed in the multiscale model can be attenuated by changing the patch recomposition strategy (using only the central value instead of averaging all values). However, doing so may also produce other kind of undesirable artifacts (like staircasing effects as in the up right example). In order to see the change, the reader is invited to zoom on these images (with a viewer that performs NN interpolation; otherwise other filtering procedure may attenuate the grain).*

4.3. Comparison with iterated NN projections. In this paragraph, we compare the multiscale model with the model obtained by iterating NN projections at each scale (which is a simplification of the model of Kwatra et al. [28] with only 3×3 patches and working only with the ℓ^2 distance). As one can observe in Fig. 13, in terms of visual proximity of the generated textures, the multiscale iterated NN do not perform better than the multiscale OT.

However, it is clear that the multiscale OT does not optimize the same patch-based cost function. Indeed, in these figures, we also plotted the following NN energy at each scale

$$E^{\ell}_{\mathrm{NN}} = \sum_{x} \left\| V^{\ell}(x + 2^{\ell}\omega) - \mathrm{NN}_{u^{\ell}}\left(V^{\ell}_{x+2^{\ell}\omega}\right) \right\|^2 \tag{34}$$

where we sum the square distances from each 3×3 patch of the current synthesis V^{ℓ} at scale ℓ to its nearest neighbor in u^{ℓ}. Of course the E_{NN} values obtained with iterated NN are lower than the ones attained with the (non-iterative) OT assignment, but as we said, this does not reflect a higher fidelity in the synthesized texture.

This experiment highlights that the NN energy E^{ℓ} is not sufficient to account for the quality of the synthesized texture. In other words, it is not true that a perfect synthesis is given by any image whose patches can all be found somewhere in the input image. Indeed, a counter-example is given by one non-trivial texture u in which by chance happens a constant patch with color c; then a constant image with color c would be considered as a perfect synthesis for this energy E_{NN} but a complete failure for a human observer. Therefore, this experiment underlies the need of a statistical control in the synthesis process, as was already stated by [59] and [21]. The poor quality of the images generated by iterated NN may explain why the authors of [28] used several patch sizes at each scale and relied on a more sophisticated patch aggregation than a weighted average.

In Fig. 13 we also propose to iterate NN projections using the same subset of 1000 patches as in OT. This subsampling procedure leads to a very degenerate texture (compared to OT or iterated NN with all patches). Again, this confirms the gain of including weights in the patch comparison, especially in order to get a non-degenerate multiscale model. But this also illustrates that OT is much more robust to the subsampling of the target distribution.

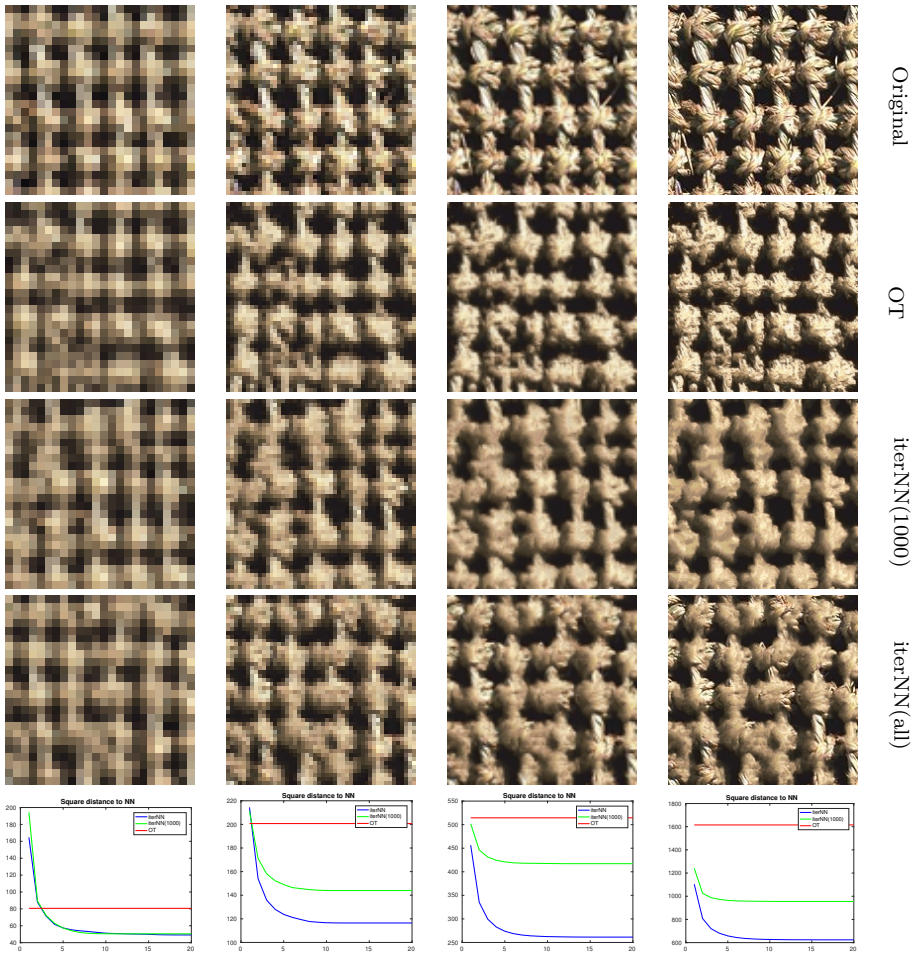

FIG. 13. ***Comparison with iterated NN projection.*** *In the multiscale framework, we compare the patch OT with iterated NN projections. The first row contains the original image at multiple scales. The second row contains the synthesis obtained with* 3×3 *patch OT at multiple scales (at each scale, the target distribution is formed with* 1000 *patches randomly chosen in the exemplar). The third row contains the synthesis obtained by iterating NN projection at each scale with the same* 1000 *patches. The same for the fourth row excepts that we perform NN projection on all the exemplar patches. In the last row we display the sum of square distances from the synthesis patches to their NN in the exemplar patches. See the text for comments.*

Finally, let us add that it does not make sense to iterate the OT assignment T_v (with same v) because the source distribution is not the same after one pass. Again, in contrast to iterated NN, these OT maps are not designed to optimize a patch-based proximity criterion.

4.4. Discussion on the model parameters. First let us recall that for the monoscale model, we propose to set the number of iterations of ASGD to 10^6 (as discussed in Section 2.3) and to sample the empirical 3×3 patch distribution of a texture with 1000 patches (as discussed in Section 4.2). These values should not be seen as parameters but as generic working values that are set as large as possible to ensure technical practicability of the corresponding algorithms (the visual results could only improve by increasing these values).

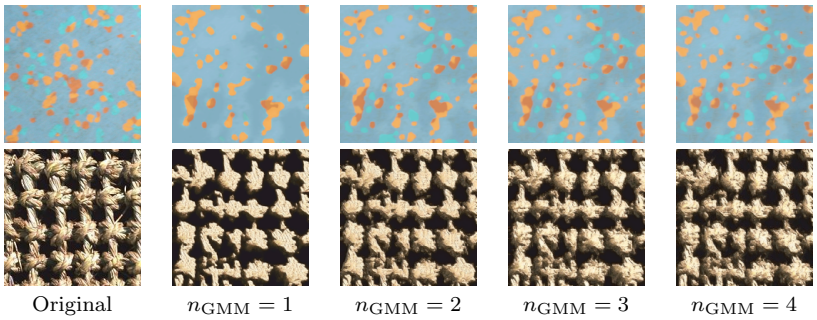

FIG. 14. ***Varying the number of Gaussian components.*** *We vary the number n_{GMM} of components of the GMM at each scale while the number of scales is fixed to 4. One can observe that we miss some parts of the synthesis when using too few components in the GMM. In general we observed that the results do not further improve when taking more than 4 components.*

The proposed multiscale model has essentially two parameters which can be easily tuned manually. The first one is the number n_{GMM} of components in the source GMM distributions. As one can see on Fig. 14, when using too few Gaussian components, the synthesis model tends to forget some perceptual components of the texture (like the blue stains of the first example). This may seem quite surprising at first because the target distribution is still the same. But this actually confirms that the transport maps obtained by ASGD have a good statistical compliancy (even if ASGD is slow to converge). Changing the source measure μ with a distribution that does not fit the input data naturally deforms the transport maps. It is thus important to use enough Gaussian components in order to properly fit the input patch distribution. As reflected by Fig. 14, increasing n_{GMM} may only increase the quality of the synthesized texture, but too large values may cause instabilities in the GMM estimation. In view of our experiments, we observed that using $n_{\mathrm{GMM}} = 4$ Gaussian components was sufficient for all the considered textures. This observation has to be related to the choice made in GMM models for denoising natural images: the authors of [70] suggest to use 200 generic Gaussian models whereas the authors of [68] use ≈ 20 Gaussian models adapted to the input image (see the discussion in [58]). Of course, in the very particular case of texture image, the number of needed Gaussian components should be much lower. Let us recall that the GMM modeling is only used during the model estimation (for computing the transport maps T^ℓ at scales $\ell > 0$), and thus the chosen value for n_{GMM} has no impact on the computational time for synthesis.

When the number of scales increases, the visual quality of the synthesized texture is not expected to worsen since we reimpose more and more statistics. This is confirmed by the results of Fig. 15. However, beyond a certain number of scales, the synthesis algorithm locally produces quasi-verbatim copy of the exemplar, because there is not enough patch variability at coarse scale. This can be observed on the first example of Fig. 15: with an exemplar of size 256×256, the exemplar at scale 6 is of size 4×4 and thus only contains 4 patches of size 3×3.

Of course, the main advantage to increase the number of scales is to synthesize larger and larger geometric structures. It is thus reasonable to set the number of scales as the smallest L such that the patch domain at scale $L-1$ (namely $2^{L-1}\omega$) is large enough to contain the geometric elements of the exemplar. Ideally, the Gaussian model should fit the exemplar at the coarsest scale, and all the applied local maps should be enough to capture the non-Gaussian behavior contained in the fine scales.

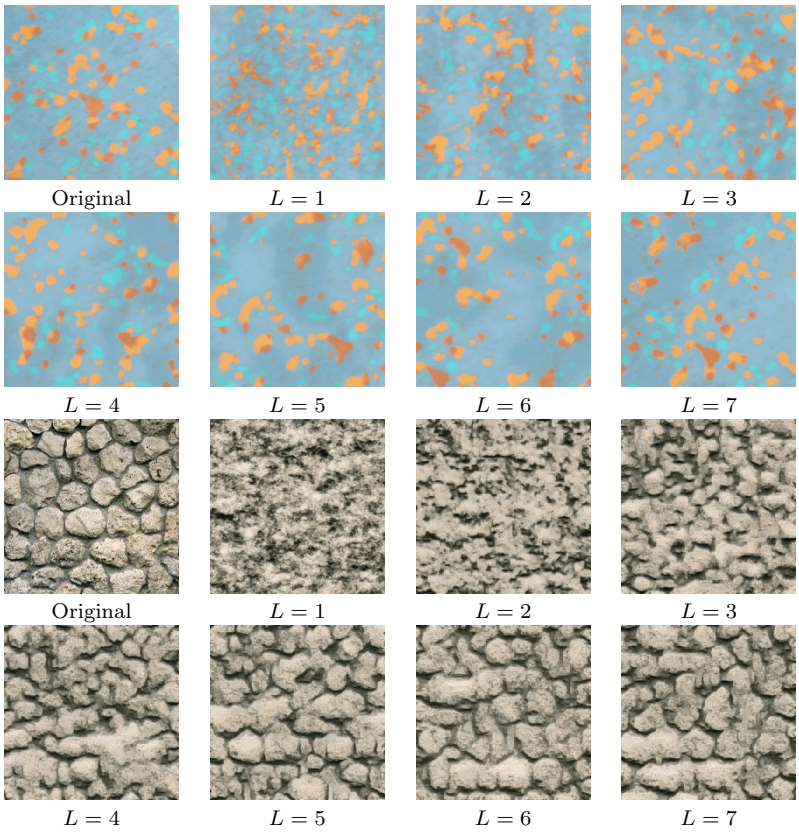

FIG. 15. ***Varying the number of scales.*** *We vary the number L of scales while the number of Gaussian components is fixed to 4. Increasing the number of scales allows to reproduce larger geometric structures of the exemplar. But using too many scales pushes the model to use very large parts of the exemplar.*

Generally speaking, for very complex textures (which contain structured information at all scales), the optimal number of scales is related to the size of the input original image (essentially proportional to the $\log_2$ of the dimensions). During the preparation of this paper, we often used 4 scales for small textures (of size ≤ 256) and 6 scales for large textures (of size $\geqslant 512$).

Therefore, one must set the number of scales in order to realize a compromise between recovered geometric structures and variability of the resulting model. Let us mention that several L may satisfy this criterion and one should favor the smallest one, if only because the corresponding synthesis algorithm is faster.

4.5. Comparison with other texture models. In this paragraph, we compare the texture synthesis results obtained with different models, including state of the art methods.

In Fig. 16, we compare with the models of [50] and [59]. Recall that the method by Portilla and Simoncelli [50] consists in alternate projections on several well-chosen statistical constraints (some of them corresponding to correlations of complex wavelet coefficients), starting from a white noise initialization. The method of Tartavel et al. [59], also starting from a white noise, optimizes the visual proximity to the exemplar with a balance between different criteria: the color distribution, the power spectrum, and

the sparsity of patches in an adapted dictionary. For the examples of Fig. 16, the algorithm of [59] was used with default parameters. For the larger examples shown in Fig. 17, this algorithm was used with 4 scales and 12×12 patches (so that the receptive field is the same than our method with 6 scales and 3×3 patches).

As one can observe on Fig. 16, the multiscale OT model leads to results that are similar to [59] and often better than [50]. One important difference between our model and those two other methods is that at each scale, the images are generated by averaging a few patches which are directly taken in the exemplar. Therefore, it will be very unlikely with multiscale OT to create false colors which do not appear in the exemplar. Even if such false colors artifacts may be critical for the human evaluation of success/failure of texture synthesis, it must be said that taking patches directly in the exemplar poses an undeniable limitation in the innovation. However, even with this constraint, the results of Fig. 16 show that the multiscale OT has the capacity to generate innovative content while always being locally close to the exemplar.

Again, on several examples of Fig. 16, one can observe a slight loss of textural grain in the result of multiscale OT. In comparison the methods [50] and [59] better respect the grain in the fine scales, because of the autocorrelation constraint in [50] and of the power spectrum term in [59] (particularly with the final post-processing steps of "histogram" and "spectrum" transfer). But the price to pay with these methods is to observe some local oscillations everywhere while the multiscale OT leads to cleaner local geometric structures. Finally let us observe that for some complex textures (like the "chalks" texture, Fig. 16, fifth row), the method [59] may fail because sparse representations in a dictionary are not able to account for the variability in the patches, while the OT-based method is inherently designed to respect the patch variability at several scales.

In Fig. 17, we provide other comparative results with very large textures: the exemplar textures have size 512×512 while the synthesized images have size 512×512 or 1024×1024. In addition to the methods of [50] and [59] that we already discussed above, we compare with the patch-based method of [52] and the neural network method of [18]. The model by Raad et al. [52] consists in progressive sampling of the texture using local conditional Gaussian models estimated by a set of similar patches taken in the exemplar. Besides, as in our model, the synthesis is performed in a multiscale fashion: the synthesis at one scale impacts the selection of similar patches at the adjacent finer scale. Thus, this model [52] can be thought of as a multiscale and randomized extension of [13]. The synthesis algorithm by Gatys et al. [18] follows the same statistical framework than Portilla and Simoncelli [50] except that they use second-order statistics extracted at each layer of a convolutional neural network (more precisely spatially averaged Gram matrices of the responses).

Fig. 17 confirms the need of a constraint on global statistics of the synthesized texture. Indeed, several failure examples of the patch-based method [52] can be explained by the fact that progressive sampling does not ensure any global statistical compliancy, as was already reflected by the "growing garbage" effect observed in the original method by Efros and Leung [13]. This seems to be in contradiction with the result by Levina and Bickel [32] who showed a statistical consistency result for such an algorithm; but let us emphasize that this theoretical result holds true in an asymptotic setting where the dimension of the exemplar texture grows to infinity. One benefit of the multiscale OT model is that it is designed to enforce such statistical consistency with a fixed-size exemplar. Nevertheless, as can be observed in [52], the model by Raad et al. is very adapted to synthesis of quasi-periodic textures, for which our model will often fail (recall Fig. 10, fifth row).

Let us now compare to the method by Gatys et al. [18]. We observed that for many structured textures, the results provided by Gatys et al. are nearly perfect in the sense that the synthesized content locally resembles parts from the exemplar while being always slightly different. In other words, this method precisely respects the geometric structures while bringing enough innovation; this is clear that the OT-based method does not respect the local geometry in such a precise way. However, on the first example of Fig. 17, one can observe a drift in the color distribution in the result of [18]. And indeed, it is not obvious to understand the link between the color distribution and the CNN responses used in [18]. In contrast, the statistical guarantee provided by OT allows to avoid such artifacts. This is also true for other examples of Fig. 17 but in a less obvious manner. Moreover, we encourage the reader to observe the results of Fig. 17 at different scales (i.e. with several zoom-in factors, or by varying the distance to the screen used for display). In particular, one may observe that the method of [18] does not have the same behavior on very large scale structures: in the two last examples of Fig. 17, one can observe that multiscale OT is better able to retrieve medium-range correlations than the CNN-based method. This reflects that the power spectrum of the texture is not directly taken into account in the CNN statistics, while it guides the coarse-scale synthesis in the multiscale OT model thanks to the Gaussian initialization.

5. Conclusion. In this paper we introduced a texture model based on local transformations of Gaussian random fields using semi-discrete OT maps in the 3×3 patch space. Such OT maps allow to reimpose the patch distribution of the exemplar texture in a non-parametric way. In addition to strong theoretical guarantees (stationarity, long-range independence) such random fields inherit the medium-range correlations (power spectrum) from the Gaussian model while exhibiting sharper geometric details. Once estimated, these semi-discrete OT maps can be written as weighted nearest-neighbor assignments which can be applied to all patches in parallel.

We also proposed a multiscale extension of this model that allows to reimpose the patch distribution at different scales. This multiscale OT model is able to synthesize structured textures in a very efficient manner. Except on some textures with very constrained local geometry, the visual results are better than state of the art methods while being much faster. In particular, we demonstrated that applying one single patch OT at each scale is both faster and statistically more relevant than using iterated nearest neighbor projections as in [28]. Several synthesis results demonstrated the benefit of imposing a global statistical control at several scales. Let us also emphasize on the fact that the multiscale OT model only applies a series of weighted nearest neighbor projections to a well-chosen Gaussian field. All of these elementary steps are quite simple to understand and thus the results (and in particular the failure cases) of this model can be explained easily.

The main limitation of this model is the difficulty of modelling complex geometrical constraints between patches. One way to better impose complex local geometry is to work with larger patches as is done in many texture synthesis methods, but of course copying large patches often leads to verbatim copy. It is remarkable that the multiscale OT model is able to generate new structures by using only average recomposition of very small patches. But using larger patches in this model is for now prohibited: on the one hand, the stochastic optimization framework would not be efficient enough to approximate the OT map, and on the other hand, the average recomposition of patches would introduce too much blur in the synthesis.

It is likely that this OT model could be improved by using better geometric models for the patch space. First, one can hope to build a distance that better accounts for the patch deformations, which would allow to perform local patch averages in a more relevant way than the ℓ^2 distance. Besides, such a framework could help to fit a low dimensional model for the patches extracted from an exemplar texture. This would allow to consider larger patches while keeping a reasonably low number of intrinsic dimensions. For this reason, one would hope to get adapted tools for stochastic OT that better scale with the size of patches. Yet, it seems to us that finding a patch model that is well-correlated to local human perception is a challenging issue: such a model would be at least as complex as perceptual models for color space (because the color distribution is a marginal of the patch distribution).

Finally, let us conclude on the fact that there probably exists a true limit in what can be modeled through patch statistics. On the one hand, considering too many statistics may probably force to do verbatim copy. On the other hand, if one allows degrees of freedom apart from certain statistics, one must accept the fact that the synthesis may not have everywhere the same aspect than the original (for example sampling from an absolutely continuous distribution in a Euclidean color space will most certainly generate false colors). For those reasons, the quality of a texture model should not be measured only through the visual proximity of synthesized textures to the exemplar but also through the scope of available theoretical properties, which may help to better understand its practical behavior or to use it for other kind of applications.

6. Acknowledgments. This work has been partially funded by the project TexTo of GdR ISIS. We gratefully thank the authors of [50, 59, 18, 51] for sharing their experiments and source codes. Also, we would like to thank Andrés Almansa, Yann Gousseau, Lionel Moisan, Jean-Michel Morel, Lara Raad, and Alain Trouvé for inspiring questions and discussions.

Appendix A. Convergence of Averaged Stochastic Gradient Descent.

Here we give a proof of the convergence of the stochastic gradient descent algorithm used to compute the solution to the semi-discrete optimal transport problem. This proof is already given in the supplementary material of [44] but we emphasize on the fact that differentiability is not required everywhere.

Let $\mathcal{H}$ be a Hilbert space, and $f : \mathcal{H} \to \mathbb{R}$ be a convex differentiable function that we wish to minimize. Let $(\mathcal{F}_n)$ a filtration and for each n, let $f_n : \mathcal{H} \to \mathbb{R}$ be a convex random function which is $\mathcal{F}_n$-measurable. We assume that for each $\theta \in \mathcal{H}$, f_n is differentiable at θ with $\mathbb{E}[\|\nabla f_n(\theta)\|] < \infty$ and with

$$\mathbb{E}[\nabla f_n(\theta)|\mathcal{F}_{n-1}] = \nabla f(\theta). \tag{35}$$

We also assume a uniform bound on the gradients, i.e. there exists $B > 0$ such for all $\theta \in \mathcal{H}$, $\|\nabla f_n(\theta)\| \leq B$ almost surely (this implies $\mathbb{E}[\|\nabla f_n(\theta)\|] < \infty$).

Let θ_0 be an $\mathcal{F}_0$-measurable random variable in $\mathcal{H}$, and consider the recursion

$$\theta_n = \theta_{n-1} - \gamma_n \nabla f_n(\theta_{n-1})$$
$$\bar{\theta}_n = \frac{1}{n}\sum_{k=0}^{n-1} \theta_k$$

where $\gamma_n > 0$ is a non-increasing sequence of gradient steps.

THEOREM 6 ([44]). *Let $\theta^* \in \operatorname{argmin} f$. Then*

$$\mathbb{E}[f(\bar{\theta}_n) - f(\theta^*)] \leqslant \frac{1}{2n}\left(\frac{D_n}{\gamma_n} + B^2 \sum_{k=1}^{n} \gamma_k\right),$$

where $D_n = \mathbb{E}[\|\theta_0 - \theta^*\|^2] + B^2 \sum_{k=1}^{n} \gamma_k^2$.

For example, if $\gamma_n = \frac{c}{\sqrt{n}}$ with $c > 0$, then

$$D_n = \delta_0 + B^2 c^2 \sum_{k=1}^{n} \frac{1}{k} \leqslant \delta_0 + B^2 c^2 (1 + \log n)$$

$$\sum_{k=1}^{n} \gamma_k = \sum_{k=1}^{n} \frac{1}{\sqrt{k}} \leqslant 2\sqrt{n}$$

so that

$$\mathbb{E}[f(\bar{\theta}_n) - f(\theta^*)] \leqslant \frac{1}{2\sqrt{n}}\Big(\frac{\delta_0}{c} + B^2 c(3 + \log n)\Big) = \mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right).$$

Proof. With the definition of the recursion, we can write

$$\|\theta_k - \theta^*\|^2 = \|\theta_{k-1} - \theta^*\|^2 - 2\gamma_k \langle f_k'(\theta_{k-1}), \theta_{k-1} - \theta^* \rangle + \gamma_k^2 \|f_k'(\theta_{k-1})\|^2.$$

Using the hypothesis (35) on ∇f_n leads to

$$\mathbb{E}\big[\|\theta_k - \theta^*\|^2 | \mathcal{F}_{k-1}\big] = \|\theta_{k-1} - \theta^*\|^2 - 2\gamma_k \langle \nabla f(\theta_{k-1}), \theta_{k-1} - \theta^* \rangle + \gamma_k^2 \mathbb{E}\big[\|f_k'(\theta_{k-1})\|^2 | \mathcal{F}_{k-1}\big].$$

Denoting $\delta_k = \mathbb{E}\big[\|\theta_k - \theta^*\|^2\big]$ and taking expectation leads to

$$\delta_k \leqslant \delta_{k-1} - 2\gamma_k \mathbb{E}\big[\langle \nabla f(\theta_{k-1}), \theta_{k-1} - \theta^* \rangle\big] + \gamma_k^2 \mathbb{E}\big[\|f_k'(\theta_{k-1})\|^2\big].$$

Using the bound on the gradients, we get

$$2\gamma_k \mathbb{E}\big[\langle \nabla f(\theta_{k-1}), \theta_{k-1} - \theta^* \rangle\big] \leqslant \delta_{k-1} - \delta_k + \gamma_k^2 B^2. \tag{36}$$

Besides, the convexity of f gives that

$$f(\theta^*) \geqslant f(\theta_{k-1}) + \langle \nabla f(\theta_{k-1}), \theta^* - \theta_{k-1} \rangle$$

so that

$$\langle \nabla f(\theta_{k-1}), \theta_{k-1} - \theta^* \rangle \geqslant f(\theta_{k-1}) - f(\theta^*) \geqslant 0. \tag{37}$$

Therefore $\delta_k \leqslant \delta_{k-1} + B^2 \gamma_k^2$ and recursively

$$\delta_n \leqslant \delta_0 + B^2 \sum_{k=1}^{n} \gamma_k^2 =: D_n.$$

Using convexity and summing (37) leads to

$$f(\bar{\theta}_n) - f(\theta^*) \leqslant \frac{1}{n} \sum_{k=0}^{n-1} \big(f(\theta_k) - f(\theta^*)\big) \leqslant \frac{1}{n} \sum_{k=0}^{n-1} \langle \nabla f(\theta_k), \theta_k - \theta^* \rangle.$$

Taking expectation and using again (36) gives

$$\mathbb{E}\big[f(\bar{\theta}_n) - f(\theta^*)\big] \leqslant \frac{1}{2n}\sum_{k=1}^{n}\left(\frac{\delta_{k-1}-\delta_k}{\gamma_k} + \gamma_k B^2\right).$$

Finally, since we have $\delta_k \leqslant D_k \leqslant D_n$ for $k \leqslant n$ and $\gamma_{k+1} \leqslant \gamma_k$, we get

$$\begin{aligned}\sum_{k=1}^{n}\frac{\delta_{k-1}-\delta_k}{\gamma_k} &= \frac{\delta_0}{\gamma_1} + \sum_{k=1}^{n-1}\delta_k\Big(\frac{1}{\gamma_{k+1}} - \frac{1}{\gamma_k}\Big) - \frac{\delta_n}{\gamma_n}\\ &\leqslant D_n\left(\frac{1}{\gamma_1} + \sum_{k=1}^{n-1}\Big(\frac{1}{\gamma_{k+1}} - \frac{1}{\gamma_k}\Big)\right) - \frac{\delta_n}{\gamma_n}\\ &= \frac{D_n}{\gamma_n} - \frac{\delta_n}{\gamma_n} \leqslant \frac{D_n}{\gamma_n}.\end{aligned}$$

Plugging that in the last inequality gives the desired bound. ∎

REFERENCES

[1] C. Aguerrebere, Y. Gousseau, and G. Tartavel, *Exemplar-based Texture Synthesis: the Efros-Leung Algorithm*, IPOL J. Image Process. Online, 2013 (2013), pp. 213–231, http://www.ipol.im/pub/art/2013/59/.

[2] F. Aurenhammer, F. Hoffmann, and B. Aronov, *Minkowski-type theorems and least-squares clustering*, Algorithmica, 20 (1998), pp. 61–76.

[3] G. Berger and R. Memisevic, *Incorporating long-range consistency in CNN-based texture generation*, in Proc. of ICLR, 2017.

[4] J. Bruna and S. Mallat, *Audio texture synthesis with scattering moments*, arXiv preprint arXiv:1311.0407, (2013).

[5] R. Chellappa, *Two-dimensional discrete Gaussian Markov random field models for image processing*, Progress in Pattern Recognition, 2 (1985), pp. 79–112.

[6] R. Chellappa and S. Chatterjee, *Classification of textures using Gaussian Markov random fields*, IEEE Transactions on Acoustics, Speech and Signal Processing, 33 (1985), pp. 959–963.

[7] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Application*, Academic Press Inc, 1992.

[8] R. Chellappa and R. Kashyap, *Texture synthesis using 2D noncausal autoregressive models*, IEEE Transactions on Acoustics, Speech and Signal Processing, 33 (1985), pp. 194–203.

[9] G. Cross and A. Jain, *Markov random field texture models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 5 (1983), pp. 25–39.

[10] H. Derin and H. Elliott, *Modeling and segmentation of noisy and textured images using Gibbs random fields*, IEEE Transactions on pattern analysis and machine intelligence, (1987), pp. 39–55.

[11] Y. Dong, S. Lefebvre, X. Tong, and G. Drettakis, *Lazy solid texture synthesis*, in Computer Graphics Forum, vol. 27, 2008, pp. 1165–1174.

[12] A. Efros and W. Freeman, *Image quilting for texture synthesis and transfer*, ACM TOG, (2001), pp. 341–346.

[13] A. A. Efros and T. K. Leung, *Texture synthesis by non-parametric sampling*, in Proc. of the Int. Conf. on Computer Vision (ICCV), 1999, p. 1033.

[14] B. Galerne, Y. Gousseau, and J.-M. Morel, *Random phase textures: Theory and synthesis*, IEEE Trans. Image Process., 20 (2011), pp. 257 – 267.

[15] B. Galerne, A. Leclaire, and L. Moisan, *A texton for fast and flexible Gaussian texture synthesis*, in Proc. of European Signal Processing Conference (EUSIPCO), 2014, pp. 1686–1690.

[16] B. Galerne, A. Leclaire, and L. Moisan, *Texton noise*, Computer Graphics Forum, (2017).

[17] B. Galerne, A. Leclaire, and J. Rabin, *Semi-Discrete Optimal Transport in Patch Space for Enriching Gaussian Textures*, in Proc. of Geometric Science of Information (GSI), 2017, pp. 1686–1690.

[18] L. Gatys, A. S. Ecker, and M. Bethge, *Texture synthesis using convolutional neural networks*, in Advances in Neural Information Processing Systems, 2015, pp. 262–270.
[19] S. Geman and C. Graffigne, *Markov random field image models and their applications to computer vision*, in Proc. of the Int. Congress of Mathematicians, vol. 1, 1986, p. 2.
[20] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, *Stochastic optimization for large-scale optimal transport*, in Proc. of Neural Information Processing Systems (NIPS), 2016, pp. 3432–3440.
[21] J. Gutierrez, B. Galerne, J. Rabin, and T. Hurtut, *Optimal patch assignment for statistically constrained texture synthesis*, in Proc. of Scale Space and Variational Methods in Computer Vision (SSVM), 2017.
[22] C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun, *Multiscale texture synthesis*, ACM Transactions on Graphics, 27 (2008).
[23] J. Han, K. Zhou, L.-Y. Wei, M. Gong, H. Bao, X. Zhang, and B. Guo, *Fast example-based surface texture synthesis via discrete optimization*, The Visual Computer, 22 (2006), pp. 918–925.
[24] D. J. Heeger and J. R. Bergen, *Pyramid-based texture analysis/synthesis*, in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM, 1995, pp. 229–238.
[25] B. Julesz, *Visual pattern discrimination*, IRE transactions on Information Theory, 8 (1962), pp. 84–92.
[26] B. Julesz, *Textons, the elements of texture perception, and their interactions*, Nature, 290 (1981), p. 91.
[27] J. Kitagawa, Q. Mérigot, and B. Thibert, *A Newton algorithm for semi-discrete optimal transport*, Journal of the European Math Society, (2017).
[28] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, *Texture optimization for example-based synthesis*, ACM TOG, 24 (2005), pp. 795–802.
[29] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, *Graphcut textures: Image and video synthesis using graph cuts*, ACM TOG, 22 (2003), pp. 277–286.
[30] C. Lantuéjoul, *Geostatistical Simulation: Models and Algorithms*, Springer, 2002.
[31] S. Lefebvre and H. Hoppe, *Parallel controllable texture synthesis*, ACM TOG, 24 (2005), pp. 777–786.
[32] E. Levina and P. Bickel, *Texture synthesis and nonparametric resampling of random fields*, The Annals of Statistics, 34 (2006), pp. 1751–1773.
[33] B. Lévy, *A numerical algorithm for L2 semi-discrete optimal transport in 3D*, ESAIM: M2AN, 49 (2015), pp. 1693–1715.
[34] J. Lewis, *Texture synthesis for digital painting*, in Proceedings of the Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, ACM, 1984, pp. 245–252.
[35] J. Lewis, *Algorithms for solid noise synthesis*, in Proceedings of the Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, ACM, 1989, pp. 263–270.
[36] C. Li and M. Wand, *Combining Markov random fields and convolutional neural networks for image synthesis*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2479–2486.
[37] M. Li, C.and Wand, *Precomputed real-time texture synthesis with Markovian generative adversarial networks*, in European Conference on Computer Vision, Springer, 2016, pp. 702–716.
[38] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, *Real-time texture synthesis by patch-based sampling*, ACM TOG, 20 (2001), pp. 127–150.
[39] A. Lippman, *Maximum entropy method for expert system construction*, PhD thesis, Brown Univ., Providence, RI (USA), 1986.
[40] G. Liu, Y. Gousseau, and G. Xia, *Texture synthesis through convolutional neural networks and spectrum constraints*, in Int. Conf. on Pattern Recognition (ICPR), IEEE, 2016, pp. 3234–3239.
[41] Y. Lu, S.-C. Zhu, and Y. N. Wu, *Learning frame models using cnn filters*, in 31th conference on artificial intelligence, 2016.
[42] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382, John Wiley & Sons, 2007.
[43] Q. Mérigot, *A multiscale approach to optimal transport*, Computer Graphics Forum, 30 (2011), pp. 1583–1592.
[44] E. Moulines and F. Bach, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning*, in Proc. of Neural Information Processing Systems (NIPS), 2011, pp. 451–459.
[45] D. Mumford and A. Desolneux, *Pattern Theory: The Stochastic Analysis of Real-World Signals*, A K Peters/CRC Press, 2010.

[46] R. Paget and I. D. Longstaff, *Texture synthesis via a noncausal nonparametric multiscale markov random field*, IEEE transactions on image processing, 7 (1998), pp. 925–931.
[47] G. Peyré, *Sparse modeling of textures*, Journal of Mathematical Imaging and Vision, 34 (2009), pp. 17–31.
[48] K. Popat and R. Picard, *Novel cluster-based probability model for texture synthesis, classification, and compression*, in Visual Communications and Image Processing, vol. 2094, International Society for Optics and Photonics, 1993, pp. 756–769.
[49] J. Portilla and E. Simoncelli, *Representation and synthesis of visual texture*, http://www.cns.nyu.edu/~lcv/texture/.
[50] J. Portilla and E. Simoncelli, *A parametric texture model based on joint statistics of complex wavelet coefficients*, Int. J. Comput. Vis., 40 (2000), pp. 49–70.
[51] L. Raad, A. Davy, A. Desolneux, and J.-M. Morel, *A survey of exemplar-based texture synthesis*, arXiv preprint arXiv:1707.07184, (2017).
[52] L. Raad, A. Desolneux, and J. Morel, *A Conditional Multiscale Locally Gaussian Texture Synthesis Algorithm*, J. Math. Imaging Vision, 56 (2016), pp. 260–279.
[53] J. Rabin, J. Delon, and Y. Gousseau, *A statistical approach to the matching of local features*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 931–958.
[54] J. Rabin, J. Delon, and Y. Gousseau, *Removing artefacts from color and contrast modifications*, IEEE Trans. Image Process., 20 (2011), pp. 3073–3085.
[55] J. Rabin, G. Peyré, J. Delon, and M. Bernot, *Wasserstein barycenter and its application to texture mixing*, in Proc. of Scale Space and Variational Methods in Computer Vision (SSVM), 2012, pp. 435–446.
[56] F. Santambrogio, *Optimal transport for applied mathematicians*, Birkuser, NY, (2015).
[57] O. Sendik and D. Cohen-Or, *Deep correlations for texture synthesis*, ACM Transactions on Graphics (TOG), 36 (2017), p. 161.
[58] S. Tabti, *Modélisation des images par patchs pour leur restauration et leur interprétation. Applications à l'imagerie SAR.*, PhD thesis, Télécom ParisTech, 2016.
[59] G. Tartavel, Y. Gousseau, and G. Peyré, *Variational texture synthesis with sparsity and spectrum constraints*, Journal of Mathematical Imaging and Vision, 52 (2015), pp. 124–144.
[60] G. Tartavel, G. Peyré, and Y. Gousseau, *Wasserstein Loss for Image Synthesis and Restoration*, SIAM J. on Imaging Sciences, 9 (2016), pp. 1726–1755.
[61] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, *Texture networks: feed-forward synthesis of textures and stylized images*, in Proc. of the Int. Conf. on Machine Learning, vol. 48, 2016, pp. 1349–1357.
[62] J. J. Van Wijk, *Spot noise texture synthesis for data visualization*, in ACM SIGGRAPH Computer Graphics, vol. 25, 1991, pp. 309–318.
[63] M. Varma and A. Zisserman, *Texture classification: are filter banks necessary?*, in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2003.
[64] C. Villani, *Topics in Optimal Transportation*, American Math. Society, 2003.
[65] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, *State of the art in example-based texture synthesis*, in Eurographics, State of the Art Reports, 2009, pp. 93–117.
[66] L.-Y. Wei and M. Levoy, *Fast texture synthesis using tree-structured vector quantization*, in Proc. SIGGRAPH '00, 2000, pp. 479–488.
[67] G. Xia, S. Ferradans, G. Peyré, and J. Aujol, *Synthesizing and Mixing Stationary Gaussian Texture Models*, SIAM J. on Imaging Sciences, 7 (2014), pp. 476–508.
[68] G. Yu, G. Sapiro, and S. Mallat, *Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity*, IEEE Trans. Image Process., 21 (2012), pp. 2481–2499.
[69] S. Zhu, Y. Wu, and D. Mumford, *Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling*, Int. J. Comput. Vis., 27 (1998), pp. 107–126.
[70] D. Zoran and Y. Weiss, *From learning models of natural image patches to whole image restoration*, in Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), 2011, pp. 479–486.

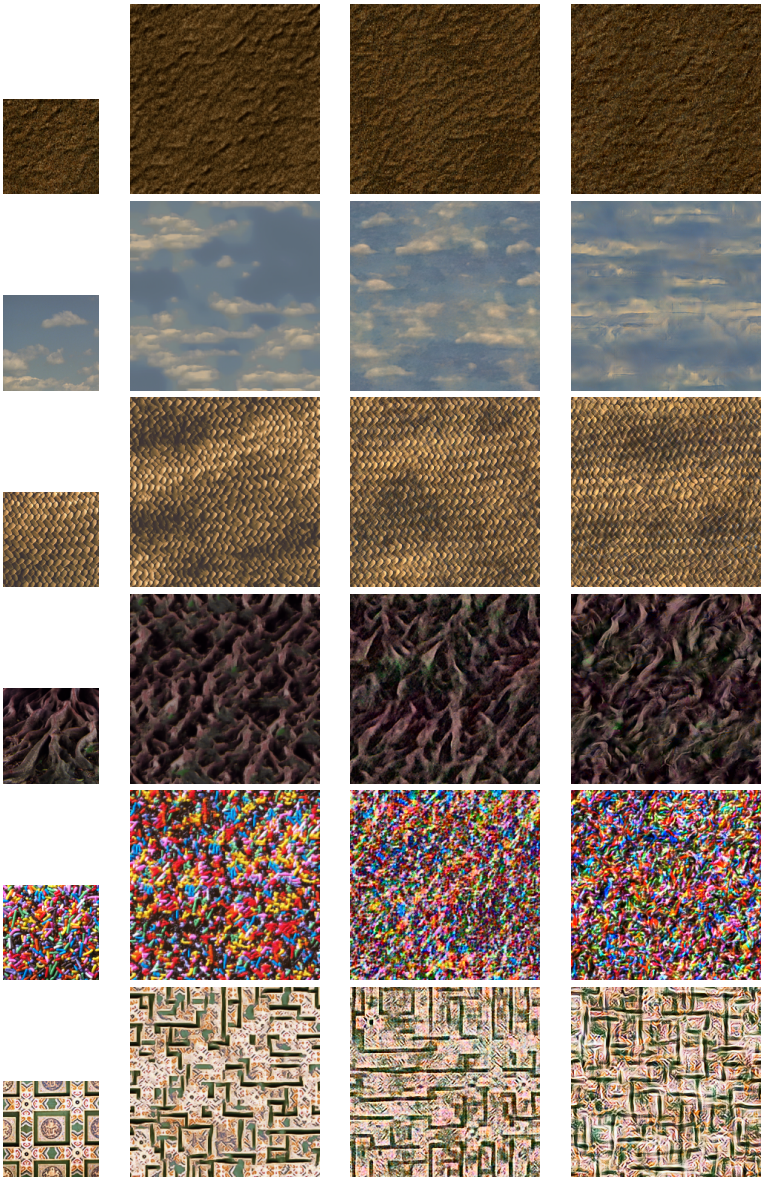

FIG. 16. ***Comparison.*** *In this figure we compare several models for texture synthesis. In the first column, we display the exemplar textures (size* 128×128*). In the other columns, we display the corresponding synthesized textures (size* 256×256*) obtained with the multiscale OT model (2nd column), the method of Tartavel et al. [59] (2nd column), and the method of Portilla and Simoncelli [50] (abbreviated P. and S.). The images of column* 3 *were generated with the implementation provided by the authors of [59]. The images of column* 4 *were generated by the reference implementation [49].*

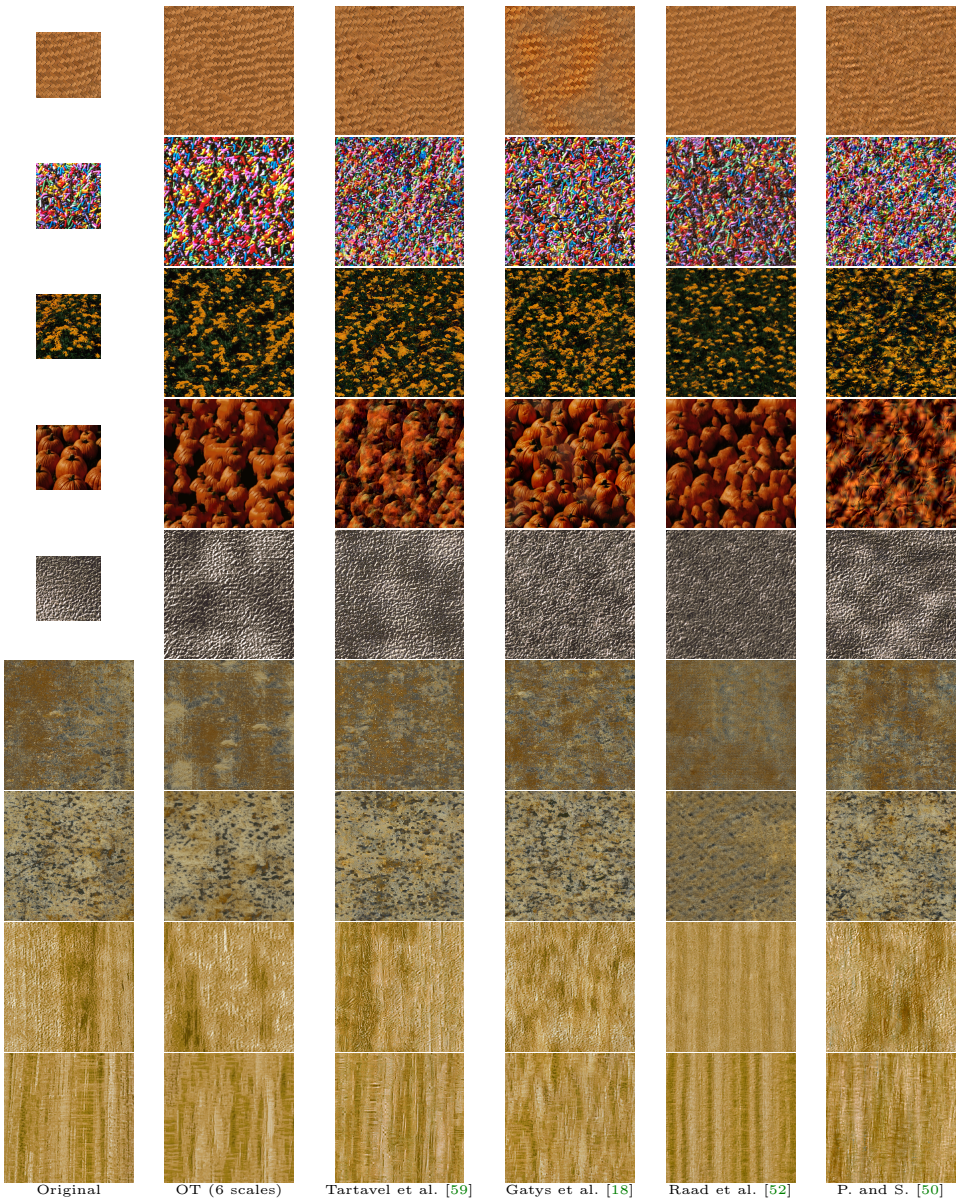

FIG. 17. ***Comparison.*** *In this figure we compare several models for texture synthesis. In the first column, we display the exemplar textures (size* 512×512*). In the other columns, we display the corresponding synthesized textures (size* 1024×1024*) obtained with the multiscale OT model (2nd column), the method of Tartavel et al. [59] (3rd column), the method of Gatys et al. [18] (4th column), the method of Raad et al. [52] (5th column), the method of Portilla and Simoncelli [50] (abbreviated P. and S.) (6th column). The images of columns* $4, 5, 6$ *were generated by the authors of [51]. The images of column* 3 *were generated with the implementation provided by the authors of [59] (with* 4 *scales and* 12×12 *patches).*