A Texton for Fast and Flexible Gaussian Texture Synthesis

Bruno Galerne, Arthur Leclaire, Lionel Moisan

A TEXTON FOR FAST AND FLEXIBLE GAUSSIAN TEXTURE SYNTHESIS

Bruno Galerne, Arthur Leclaire, Lionel Moisan

Université Paris Descartes, MAP5, CNRS UMR 8145, France

ABSTRACT

Gaussian textures can be easily simulated by convolving an initial image sample with a conveniently normalized white noise. However, this procedure is not very flexible (it does not allow for non-uniform grids in particular), and can become computationally heavy for large domains. We here propose an algorithm that summarizes a texture sample into a synthesis-oriented texton, that is, a small image for which the discrete spot noise simulation (summed and normalized randomly-shifted copies of the texton) is more efficient than the classical convolution algorithm. Using this synthesis-oriented texture summary, Gaussian textures can be generated in a faster, simpler, and more flexible way.

***Index Terms*—** Spot noise, texton, Gaussian texture, texture synthesis, error reduction algorithm

1. INTRODUCTION

Among the various existing models of textures, Gaussian textures form an interesting class, in particular because they rely on a mathematical model that is very well adapted to theoretical investigations. Gaussian textures allow for by-example synthesis [4], and can be easily generalized to dynamic scenes or used for texture mixing [10]. The classical spectral simulation, which is based on the discrete Fourier transform and consists in convolving a standard Gaussian white noise by a kernel image k, has some limitations: 1) The underlying Gaussian model is implicitly periodic; 2) It does not allow for local variations of the kernel or the grid.

A Gaussian texture can be approximated by a high-intensity discrete spot noise (DSN), obtained by summing randomly-shifted copies of the kernel k along the points of a Poisson process of intensity λ. The direct simulation of the DSN is simple and allows parallel local evaluation using standard computer graphics techniques for the Poisson process simulation [5]. Still, the DSN approximation of a Gaussian texture is satisfying only for sufficiently high intensity λ, so that the DSN simulation is generally not faster than the spectral simulation. In particular, using the compact texton introduced in [1] as a kernel for DSN synthesis generally results in a very poor approximation for small values of λ.

In this paper we show that, given an exemplar texture image u, it is possible to compute a synthesis-oriented texton (SOT) having a prescribed small support and for which the associated DSN is close to the Gaussian texture associated with u even for a low intensity λ, as illustrated in Fig. 1. This SOT is computed using the classical error reduction algorithm, introduced by [3] for phase retrieval, with a random phase initialization (Section 3). As will be shown in Section 4, for an average number of 30 impacts per pixels, the DSN associated with the SOT produces visually satisfying results, and is thus more competitive than the spectral simulation algorithm.

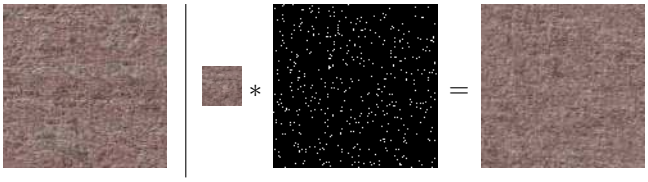

Fig. 1. **Spot noise synthesis at low intensity.** The synthesized texture on the right was obtained by the convolution of a synthesis-oriented texton with a sparse Poisson process. The exemplar texture is shown on the left.

2. SPOT NOISE TEXTURE MODELS

2.1. DSN and ADSN models on $\mathbb{Z}^2$

We first describe the DSN and ADSN models on $\mathbb{Z}^2$, since it is the most natural framework. In the following, $h : \mathbb{Z}^2 \to \mathbb{R}$ is a function with finite support S_h, and $|A|$ stands for the cardinality of a subset A of $\mathbb{Z}^2$.

The DSN on $\mathbb{Z}^2$ with spot h and intensity $\lambda > 0$, originally introduced in [9], is the stationary random process $F_{\lambda,h}$ on $\mathbb{Z}^2$ defined by

$$\forall \mathbf{x} \in \mathbb{Z}^2, \quad F_{\lambda,h}(\mathbf{x}) = \sum_{i \geq 1} h(\mathbf{x} - \mathbf{X}_i) \ ,$$

where the points $\mathbf{X}_i$ are chosen according to a Poisson point process on $\mathbb{Z}^2$ with intensity λ. If one denotes by * the convolution product on $\mathbb{R}^{\mathbb{Z}^2}$, one can see that $F_{\lambda,h} = h * P_\lambda$ where the random variables $P_\lambda(\mathbf{y}) = |\{i, \mathbf{X}_i = \mathbf{y}\}|$ are i.i.d. and follow a Poisson distribution with intensity λ.

The mean value of $F_{\lambda,h}$ is given by $m = \mathbb{E}\Big(F_{\lambda,h}(\mathbf{x})\Big) = \lambda \sum_{\mathbf{y} \in \mathbb{Z}^2} h(\mathbf{y})$, and, setting $\tilde{h}(\mathbf{x}) = h(-\mathbf{x})$, its covariance is

$C(\mathbf{v}) = \mathbb{E}\Big((F_{\lambda,h}(\mathbf{x})-m)(F_{\lambda,h}(\mathbf{x}+\mathbf{v})-m)\Big) = \lambda h * \tilde{h}(\mathbf{v})$. Notice that $\mathrm{Supp}(C) \subset S_h - S_h := \{\mathbf{x}-\mathbf{y}\ ;\ \mathbf{x},\mathbf{y} \in S_h\}$.

The renormalized DSN defined by

$$G_{\lambda,h} = \frac{F_{\lambda,h} - \mathbb{E}(F_{\lambda,h})}{\sqrt{\lambda}} = \frac{1}{\sqrt{\lambda}}\Big(h * P_\lambda - \lambda \sum_{\mathbf{y}\in\mathbb{Z}^2} h(\mathbf{y})\Big)$$

has zero-mean and covariance function $h * \tilde{h}$, and it is well-known [8] that when $\lambda \to +\infty$, $G_{\lambda,h}$ converges in distribution to the Gaussian random field G_h with same mean and covariance, which is thus called the ADSN associated to h. Notice that the random process G_h can be simply simulated by $k * W$, where W is a standard Gaussian white noise on $\mathbb{Z}^2$ and $k : \mathbb{Z}^2 \to \mathbb{R}$ is any function with compact-support such that $k * \tilde{k} = h * \tilde{h}$. Such a square root k of the covariance is called a texton in [1, 10]. In the present paper, we will call a SOT any compactly-supported function k which is an approximate square root of the covariance and such that the approximation of G_k by $G_{\lambda,k}$ is visually satisfying even for low values of λ.

In the following, the random fields $G_{\lambda,h}$ and G_h will be referred to as $\mathrm{DSN}_\lambda(h)$ and $\mathrm{ADSN}(h)$ respectively.

2.2. DSN and ADSN models on a circular finite domain

We now consider the case of a finite (circular) domain, which is a more adapted framework for numerical simulations. Let $\Theta \subset \mathbb{Z}^2$ be a finite rectangular domain of size $M \times N$, equipped with the addition modulo (M, N) and the circular convolution operator $\odot$. We will assume that Θ contains a translation of S_h, so that h can be identified to a function defined on Θ. This allows us to consider the circular DSN (CDSN) associated to h, denoted by $F^\Theta_{\lambda,h}$, which is built by adding copies of h positioned according to a Poisson point process on Θ with intensity λ. All the properties mentioned in Section 2.1 have their circular counterparts. In particular, the renormalized CDSN $G^\Theta_{\lambda,h}$ has mean 0, covariance equal to $h \odot \tilde{h}$, and when $\lambda \to +\infty$ it converges in distribution to the Gaussian random field G^Θ_h with same mean and covariance, called the circular ADSN (CADSN) associated to h. In the following, the random fields $G^\Theta_{\lambda,h}$ and G^Θ_h will also be referred to as $\mathrm{CDSN}^\Theta_\lambda(h)$ and $\mathrm{CADSN}^\Theta(h)$.

It is interesting to remark that a restriction of $\mathrm{DSN}(h)$ to a rectangular domain $\Omega \subset \mathbb{Z}^2$ can be seen as a restriction of a well-chosen CDSN. Indeed, by construction, as soon as $\Omega - S_h \subset \Theta$, the restrictions to Ω of $G^\Theta_{\lambda,h}$ and $G_{\lambda,h}$ share the same distribution, which is also true for G^Θ_h and G_h.

In the following, we will need the discrete Fourier transform (DFT) on Θ, defined, for a function $v : \Theta \to \mathbb{R}$, by

$$\widehat{v}(\boldsymbol{\xi}) = \sum_{\mathbf{x}\in\Theta} v(\mathbf{x}) \exp\left(-2i\pi\Big(\frac{x_1\xi_1}{M} + \frac{x_2\xi_2}{N}\Big)\right)$$

where $\mathbf{x} = (x_1, x_2)$ and $\boldsymbol{\xi} = (\xi_1, \xi_2)$.

2.3. Optimal transport distances between ADSN models

Among the numerous assets of Gaussian texture models is the possibility to compute the L^2 optimal transport distance (OTD) between finite-dimensional marginal distributions in terms of the covariance operators. As shown in [10], the corresponding expression becomes tractable as soon as there exists a common eigenvector basis for the covariance operators. Using for example the Fourier basis, the L^2 OTD between $\mu_0 = \mathrm{CADSN}^\Theta(h_0)$ and $\mu_1 = \mathrm{CADSN}^\Theta(h_1)$ is given by

$$d^2_{OT}(\mu_0,\mu_1) = \frac{1}{|\Theta|}\sum_{\boldsymbol{\xi}\neq 0}\Big(|\widehat{h}_0|^2 + |\widehat{h}_1|^2 - 2|\widehat{h}_0^*\widehat{h}_1|\Big)(\boldsymbol{\xi}). \quad (1)$$

This allows us to define a projection of h_1 on the set of kernels associated to the model μ_0 as a solution of

$$\operatorname*{Arg\,min}_{k,\ k\odot\tilde{k}=h_0\odot\tilde{h}_0} d_{OT}(\mathrm{CADSN}^\Theta(k), \mathrm{CADSN}^\Theta(h_1))\ .$$

One particular solution $p_{h_0}(h_1)$ can be computed by imposition of the Fourier modulus:

$$\forall \boldsymbol{\xi} \neq 0,\ \widehat{p_{h_0}(h_1)}(\boldsymbol{\xi}) = \left(\frac{\widehat{h}_0\widehat{h}_0^*\widehat{h}_1}{|\widehat{h}_0^*\widehat{h}_1|}\mathbf{1}_{\widehat{h}_0^*\widehat{h}_1\neq 0}\right)(\boldsymbol{\xi}). \quad (2)$$

Notice that $p_{h_0}(h_1)$ is defined on Θ and does not *a priori* identifies to a spot $h : \mathbb{Z}^2 \to \mathbb{R}$ with compact support. Let us mention that (1) and (2) extend to the case of color CADSN, using a componentwise (R,G,B) DFT (in that case, z^* stands for the transpose conjugate of the vector z). Note that with d channels, p_{h_0} reduces to the orthogonal projection, for each $\boldsymbol{\xi}$, of $\widehat{h}_1(\boldsymbol{\xi})$ onto the $\mathbb{C}^d$-circle $\{e^{i\varphi}\widehat{h}_0(\boldsymbol{\xi}); \varphi \in \mathbb{R}\}$, leading to a geometric interpretation of (2).

The extension of those results to a non-circular framework may be difficult. One can try for example to express the OTD between the Ω-restrictions of $\mathrm{ADSN}(h_0)$ and $\mathrm{ADSN}(h_1)$, but since the corresponding covariance operators are now only Toeplitz (and not circulant), the Fourier basis functions are no longer eigenvectors. However, the last remark of Section 2.2 shows that the OTD between the Ω-restrictions is less than $d_{OT}(\mathrm{CADSN}^\Theta(h_0), \mathrm{CADSN}^\Theta(h_1))$, for any Θ containing $\Omega - S_h$. Thus, the OTD between finite ADSN pieces is controlled by the OTD between larger circular counterparts.

2.4. Simulation on a finite domain

As explained above, the DSN (resp. ADSN) on $\mathbb{Z}^2$ or a circular domain Θ can be seen as a convolution of the spot with a Poisson point process (resp. a Gaussian white noise). In the circular framework, the convolution can be performed using the DFT. In fact, this spectral method can also be used to simulate a DSN on a non-circular finite domain by using a larger domain and using a crop as post-processing (in view of the remark of Section 2.2).

However, for a DSN with a very low intensity λ, the Poisson point process is sparse so that the convolution can be performed efficiently in spatial domain. This direct summation method, summarized in Algorithm 1, can be used for the simulation of a finite restriction of $\mathrm{DSN}_\lambda(h)$, or for the simulation of $\mathrm{CDSN}_\lambda^\Theta(h)$.

Algorithm 1: DSN simulation on a finite domain Ω

- Set $\bar{\Omega} = \Omega - S_h = \{\mathbf{x} - \mathbf{y} \; ; \; \mathbf{x} \in \Omega, \mathbf{y} \in S_h\}$.
- Draw n with Poisson distribution of intensity $\lambda|\bar{\Omega}|$.
- Draw $\mathbf{x}_1, \dots, \mathbf{x}_n$ independently and uniformly in $\bar{\Omega}$.
- $\forall \mathbf{x} \in \Omega$, $f(\mathbf{x}) := \frac{1}{\sqrt{\lambda}} \left(\sum_{i=1}^n h(\mathbf{x} - \mathbf{x}_i) - \lambda \sum h \right)$.

A simple analysis show that Algorithm 1 has a mean complexity of $\mathcal{O}(\lambda |S_h| |\Omega|)$, $\lambda |S_h|$ being the mean number of impacts per pixel, versus $\mathcal{O}(|\Omega| \log(|\Omega|))$ for the spectral method. Therefore, the direct summation method will be faster for large domains Ω or for on-demand synthesis. Notice that it can be parallelized using a grid-based simulation scheme for the Poisson point process [7].

3. A SYNTHESIS-ORIENTED TEXTON

The complexity analysis above shows that the efficiency of Algorithm 1 is closely linked to the possibility of obtaining a visually satisfying texture with a small value of λ. As we said in Introduction, the texton originally proposed in [1] is inadequate for our purpose, since the convergence of the Poisson process to the Gaussian model is particularly slow for this concentrated kernel. The object of this part is to describe an algorithm that computes a kernel $h : \mathbb{Z}^2 \to \mathbb{R}$ with support $S_h \subset S$ (S being a given finite subset of $\mathbb{Z}^2$) that leads to an efficient DSN synthesis of the Gaussian texture associated to an original texture sample $u : \Omega \to \mathbb{R}$. For the sake of clarity, we assume that Ω (a $M \times N$ rectangle of $\mathbb{Z}^2$) contains S and $S - S$ (in particular, the observation is larger than the covariance support).

3.1. Gaussian model estimation

A first question that arises is the estimation of the Gaussian model associated to u. Following [1, 4, 10], we compute the mean $m = \frac{1}{|\Omega|} \sum_{\mathbf{x} \in \Omega} u(\mathbf{x})$ and the periodic autocorrelation

$$c_u = t_u \odot \tilde{t}_u \; , \quad \text{where} \quad t_u = \frac{1}{\sqrt{|\Omega|}} (u - \bar{u}) \; . \qquad (3)$$

Even if the non-periodic nature of the observation u may bias this estimator of the covariance, it is of great practical use because its DFT is given by $\widehat{c}_u = |\widehat{t}_u|^2$, and also because it is the actual covariance of the circular Gaussian field $\mathcal{N}(0, c_u)$ on Θ, which is the CADSN associated to any kernel k such that $k \odot \tilde{k} = c_u$ (or in Fourier domain, $|\widehat{k}| = |\widehat{t}_u|$).

Now, we would like to find a kernel h with support $S_h \subset S$ such that $c_u = h * \tilde{h}$. This problem is an analog of the phase retrieval problem and may lead to multiple solutions (if h is a solution, so are $-h$, $\tilde{h}$ or $-\tilde{h}$, and in particular cases there may be other solutions, see [6]). Here, because of the constraint on S_h, there is no exact solution in general, but we can look for an approximate solution by trying to solve

$$\underset{h, \; S_h \subset S}{\mathrm{Arg\,min}} \quad d_{OT}(\mathrm{CADSN}^\Omega(h), \mathrm{CADSN}^\Omega(t_u)) \; , \qquad (4)$$

where the use of circular models is justified by the need of an explicit formula for the OTD.

3.2. Alternating projections for SOT computation

The optimization problem (4) is difficult to solve, but we can propose an approximate algorithm that has proven useful in the phase retrieval literature. Indeed, Algorithm 2 below alternates between imposition of the Fourier modulus (2) and the projection $q_S : t \mapsto t\mathbf{1}_S$ on the support constraint (recall that q_S is an orthogonal projection on a convex set).

Algorithm 2: SOT computation

- Initialization: $\widehat{t} \leftarrow \widehat{t}_u e^{i\psi}$ where ψ is a uniform random phase function, and t_u is given by (3).
- Repeat (n times) $t \leftarrow q_S(p_{t_u}(t))$.

Let us remark that if u is a realization of the random phase noise (RPN) [4] associated to a texton τ with support S, then we have exactly $|\widehat{t_u}| = |\widehat{\tau}|$ so that the problem of recovering τ from u is exactly a phase retrieval problem for which the above algorithm was already proposed in [3]. But in general, we do not know if there exists such a compactly-supported texton, and even if there is one, it might not be appropriate for DSN synthesis. So in some way, we would like to take profit of the cases where the alternating projections converges towards a compact kernel which is not an ideal solution of the phase retrieval problem.

Exploiting a remark of [6], the algorithm is initialized with a random phase function. Let us stress that this choice is very important for the SOT computation. Indeed, as will be seen in Section 4, except for particular cases, the output SOTs are not quantized, and have no salient features, complying well with the requirements for DSN synthesis. The output of Algorithm 2 can be seen as a kernel with maximally random phase under Fourier modulus and support constraints.

Finally, let us mention that Algorithm 2 can be used as is to produce SOTs for RGB textures (see Section 2.3) .

The questions of the convergence and the influence of the random initialization were raised by [6]. The study below shows that both these issues are negligible in terms of the resulting Gaussian texture. Indeed, let us analyze the behavior

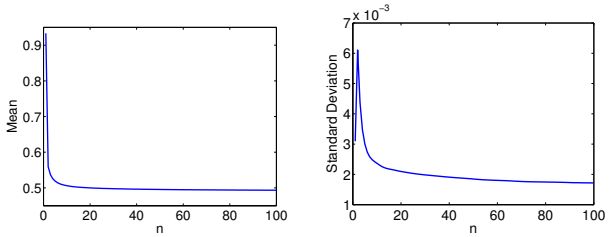

Fig. 2. **Iterates of Algorithm 2.** Evolution of the empirical mean (left) and standard deviation (right) of the RME computed after n iterations of Algorithm 2 (estimated over 1000 samples) run on a gray-level version of the texture used in the top row of Fig. 3. Observe that the mean RME quickly decreases, which means that most of the Gaussian model approximation is done in the first iterations. Notice also that the standard deviation does not tend to zero; this reflects that the algorithm does not have a unique convergence point.

of Algorithm 2 by considering the relative model error defined by

$$\mathrm{RME}(t, t_u)^2 = \frac{\sum_{\boldsymbol{\xi}\neq 0} \left(|\widehat{t_u}|^2 + |\widehat{t}|^2 - |\widehat{t_u^*}\widehat{t}|\right)(\boldsymbol{\xi})}{\sum_{\boldsymbol{\xi}\neq 0} |\widehat{t_u}|^2(\boldsymbol{\xi})}.$$

The numerator is the OTD between $\mathrm{CADSN}^{\Omega}(t_u)$ and $\mathrm{CADSN}^{\Omega}(t)$, and the denominator is the marginal variance of $\mathrm{CADSN}^{\Omega}(t_u)$.

A direct observation of the iterates shows that for each random initialization, they seem to stabilize after a small number of iterations, as already mentioned in [6]. To be more precise, we computed the empirical mean $\bar{d}_n$ and variance $\bar{\sigma}_n^2$ of the random variable $D_n = \mathrm{RME}(t_u, T_n)$, where T_n is the SOT obtained after n iterations of the algorithm with random initialization. As one can see in Fig. 2, $\bar{d}_n$ and $\bar{\sigma}_n$ do not change much for $n \geq 50$, reflecting again the quick stabilization of the iterates. Besides, the fact that $\bar{\sigma}_n$ does not tend to zero reaffirms the random nature of the output.

We also investigated the idea of running several times Algorithm 2 with different random initializations and selecting the output with the smallest RME, but numerical simulations showed that the improvement in RME (for a fixed computation time) was not significant (below 1%).

4. RESULTS

In this last part, we present experiments (see Fig. 1, 3 and 4) showing that the SOT computed by Algorithm 2 allows us to synthesize a Gaussian texture associated to a sample image u by a DSN with a very low number of impacts per pixel. For comparison purpose, we used in Fig. 4 the luminance texton t_{lum} obtained by subtracting to each channel of $\widehat{t_u}$ the phase of $\frac{1}{3}\widehat{u}_r + \frac{1}{2}\widehat{u}_g + \frac{1}{6}\widehat{u}_b$ (see [1]). Notice also that the use of small support textons may decrease the color diversity, but as mentioned in [2], it is possible to apply a simple post-processing

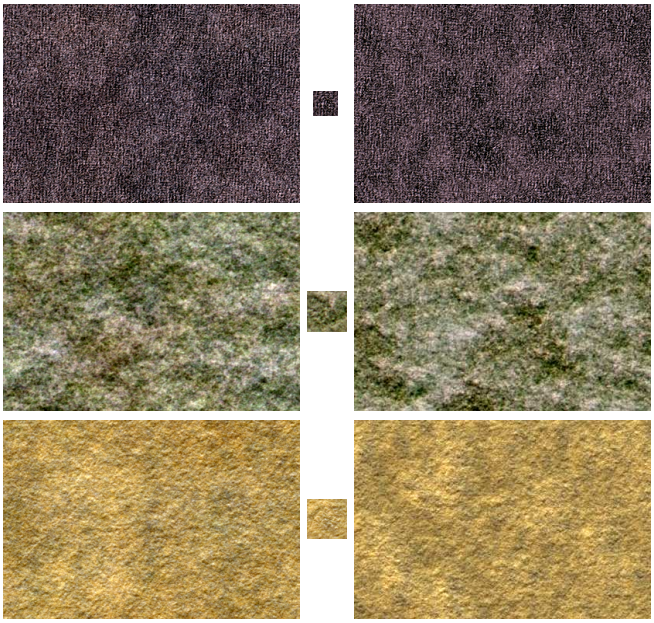

Fig. 3. **Synthesis of color textures.** One can see on the left three Gaussian textures, and on the right, the results of DSN synthesis (with 50 impacts per pixel) using the SOTs shown in the middle.

(a 3×3 linear transform of the color channels) to recover the marginal color covariance of the original sample. We applied this post-processing for the simulations of Fig. 3 only.

We can see that the DSN synthesis with the SOT is generally satisfying in terms of frequency content, even for a low number of impacts per pixel. Using the SOT, the direct summation method of Algorithm 1 thus becomes a competitive way of synthesizing Gaussian textures, with an expected number of operations per pixel below 100. As can be seen in Fig. 4, the results are as good as Gabor noise by example [5] (which requires around 1000 operations per pixel).

Concerning the precision of the model, with a reasonable intensity, the DSN synthesis is considered to be a good approximation of the ADSN. It is thus sufficient to compare the ADSNs obtained with t_u and t, both visually and using the RME. Fig. 4 shows that $\mathrm{ADSN}(t)$ and $\mathrm{ADSN}(t_u)$ have indeed a close aspect. In spite of this, the RME remains surprisingly high (we often observed RMEs between 0.4 and 0.7). Notice however that the SOT leads to lower RME value than the luminance texton cropped with the same support. It means that, although the luminance texton is a very concentrated summary of the texture, its cropped version is not the optimal way of representing the texture on a given support. The SOT is thus slightly better than the luminance texton for asymptotic synthesis, and drastically better for DSN synthesis.

To conclude, let us discuss the influence of the support size. In the Gaussian model with a circular texton of radius r, the values of two pixels at distance greater than $2r$ are in-

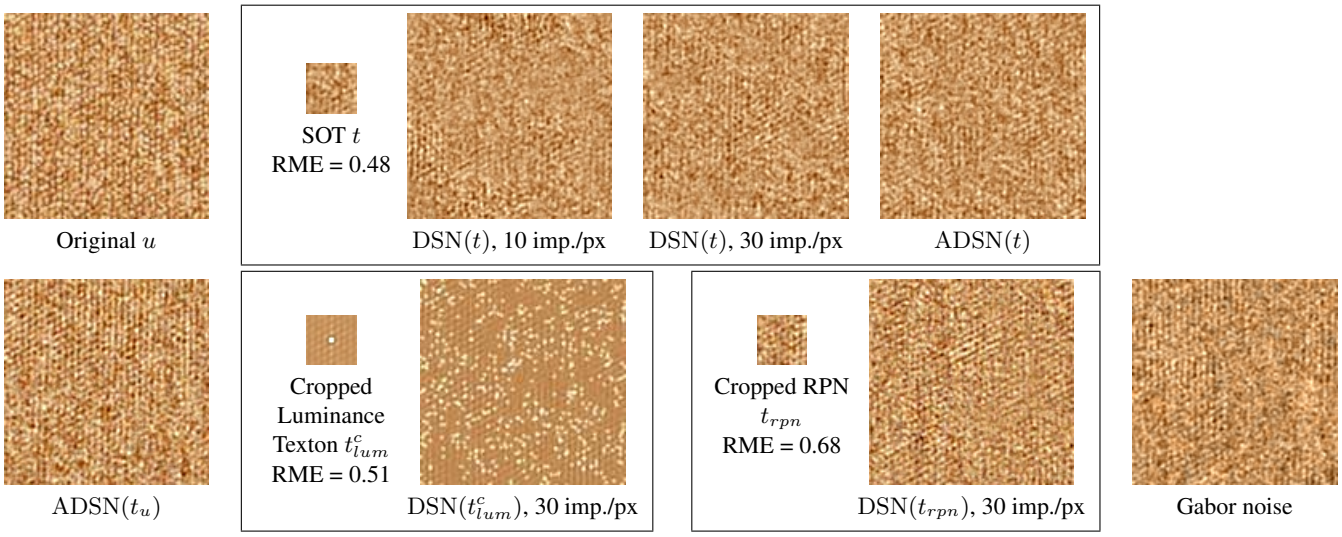

Fig. 4. **DSN synthesis of a natural color texture, comparison.** Top row: original texture (u), and the DSN and ADSN synthesis results obtained with a 31×31 SOT t computed by Algorithm 2 (1000 iterations). The DSN intensities were set in order to match a given average number of impacts per pixel. Bottom row: sample of the Gaussian model associated to u, DSN obtained with the cropped luminance texton t^c_{lum} [1], DSN obtained with a texton t_{rpn} cropped from a realization of the RPN of u (that is, the random-phase image t used in the initialization of Algorithm 2), and Gabor noise synthesis [5]. Each DSN model is displayed with its corresponding kernel. One can see that contrary to the other DSN models, the proposed SOT achieves a good visual proximity with the reference model $\mathrm{ADSN}(t_u)$ as the number of impacts per pixel attains 30. Moreover, it also defines the most accurate asymptotic model (smallest RME).

Fig. 5. **Influence of the support size.** A Gaussian texture (u), and samples of the models obtained with different SOTs t^r with circular supports of radius $r \in \{5, 15, 25\}$. As expected, the quality of approximation increases as r grows.

dependent. Therefore, increasing r results in the capture of longer-range dependencies of the original texture, as illustrated in Fig. 5. Hence, the efficiency of Algorithm 1 (and of the SOT presented here) is directly linked to the nature of the covariance of the considered Gaussian texture: in the case of very long-range dependencies, the computational speed-up may vanish and only the flexibility (on-demand synthesis, texture with local variations, non-uniform grid) remains.

REFERENCES

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

[2] A. Desolneux, L. Moisan, S. Ronsin, "A texton for Random Phase and Gaussian textures", in preparation.

[3] J.R. Fienup, "Phase retrieval algorithms: a comparison", *Applied Optics* 21(15), pp. 2758–2769, 1982.

[4] B. Galerne, Y. Gousseau, J.-M. Morel, "Random Phase Textures: Theory and Synthesis", *IEEE Trans. on Image Processing* 20(1), pp. 257–267, 2011.

[5] B. Galerne, A. Lagae, S. Lefebvre, G. Drettakis, "Gabor noise by example", *proc. SIGGRAPH* 31(4), pp. 73:1-73:9, 2012.

[6] M.H. Hayes, "The Reconstruction of a Multidimensional Sequence from the Phase or Magnitude of its Fourier Transform" *IEEE Trans. on Acoust., Speech, Signal Process.* 30(2), pp. 140–154, 1982.

[7] A. Lagae, S. Lefebvre, G. Drettakis, P. Dutré, "Procedural Noise Using Sparse Gabor Convolution", *proc. SIGGRAPH* 28(3), pp. 54:1–54:10, 2009.

[8] A. Papoulis, "High Density Shot Noise and Gaussianity", *J. Appl. Probab.* 8(1), pp. 118–127, 1971.

[9] J.J. van Wijk, "Spot noise texture synthesis for data visualization", *proc. SIGGRAPH* 25, pp. 309–318, 1991.

[10] G.-S. Xia, S. Ferradans, G. Peyré, J.-F. Aujol, "Synthesizing and Mixing Stationary Gaussian Texture Models", to appear in *SIAM J. on Imaging Science*, 2014.