HAL
open science

Analysis of a Physically Realistic Film Grain Model, and a Gaussian Film Grain Synthesis Algorithm

Alasdair Newson, Noura Faraj, Julie Delon, Bruno Galerne

To cite this version:

Alasdair Newson, Noura Faraj, Julie Delon, Bruno Galerne. Analysis of a Physically Realistic Film Grain Model, and a Gaussian Film Grain Synthesis Algorithm. 6th Conference on Scale Space and Variational Methods in Computer Vision SSVM 2017, Jun 2017, Kolding, Denmark. hal-01494123v2

HAL Id: hal-01494123

https://hal.science/hal-01494123v2

Submitted on 10 May 2017

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

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

Analysis of a Physically Realistic Film Grain Model, and a Gaussian Film Grain Synthesis Algorithm

Alasdair Newson, Noura Faraj, Julie Delon, Bruno Galerne

Laboratoire MAP5 (CNRS UMR 8145), Université Paris Descartes, Paris

Abstract. Film grain is a highly valued characteristic of analog images, thus realistic digital film grain synthesis is an important objective for many modern photographers and film-makers. We carry out a theoretical analysis of a physically realistic film grain model, based on a Boolean model, and derive expressions for the expected value and covariance of the film grain texture. We approximate these quantities using a Monte Carlo simulation, and use them to propose a film grain synthesis algorithm based on Gaussian textures. With numerical and visual experiments, we demonstrate the correctness and subjective qualities of the proposed algorithm.

Keywords: film grain, Gaussian texture, covariance, Monte Carlo simulation

1 Introduction

Film grain is the specific texture which results from the analog photographic process. This texture is highly sought after by film directors and photographers alike for its artistic qualities. Producing realistic film grain for images is therefore a crucial goal. Proposed film grain synthesis methods often rely on scanned example of film grain which is either directly blended with a given image [1,4,7], used to generate a film grain model [6,10,16,12]. Although, simple and fast, these approaches rely completely on the resolution and quality of the original scan. Furthermore, in the case where film grain is modelled as independent noise [10,16], with a variance which is dependent on the input image intensity, the grain texture is completely uncorrelated spatially, which gives a distinctly undesirable "digital" feel to the image. To tackle these issues, Newson et al [11] proposed a physically realistic model of film grain. This made use of the Boolean model [3] from the stochastic geometry literature. In the present work, we propose an approximation of the Boolean model using Gaussian textures. The central idea is to determine the covariance of the film grain produced by the Boolean model, and impose this covariance on a white noise vector. This approach has several advantages. Firstly, once the covariance matrix is known for a given image, the grain can be re-synthesized extremely quickly, since this is done by sampling a white noise vector, and multiplying it with a (very sparse) matrix. This differs from

many grain synthesis algorithms which blend a fixed scan of film grain with an image. Secondly, the texture model is based on the covariance of the physically-motivated Boolean model, which means that the parameters are meaningful. From a technical point of view, for any given couple of input pixels, we determine the covariance and the expected value of the filtered Boolean model using a Monte Carlo simulation. We impose these characteristics on a Gaussian white noise to produce the output image. This paper has two main objectives. Firstly, we analyze a previously proposed, physically realistic film grain model [11], based on the Boolean model [3]. In this analysis, we consider the first and second order statistics of the grain model and propose an algorithm to determine these statistics for any given input image. These quantities are the key characteristics of film grain, therefore their theoretical calculation is of significant value. Indeed, once these quantities are known, a variety of algorithms could be proposed to exploit this information for film grain synthesis. Our second contribution is one such algorithm which simulates film grain on a given input image by imposing the previous characteristics on the output image, using a Gaussian approximation.

2 Stochastic Modelling of Film Grain

An analog film is made up of an emulsion (a gelatin) in which many microscopic silver halide crystals are suspended. These crystals are sensitive to light, which is why they are used for photography. During the exposure, when a photograph is taken, a photon may hit one of the crystals and "sensitize" it, creating a very small amount of solid silver on the crystal. The emulsion is then "developed", that is to say a chemical compound is introduced into the emulsion. This compound turns only the sensitized grains into solid silver grains, which means that the density of the grains depends on the intensity of the light (the image) which was shone upon them. A comprehensive explanation of the photographic process can be found in [9]. In order to model the previous process, Newson et al. [11] proposed to use an inhomogeneous Boolean model[1] from the stochastic geometry literature [3]. This model was also implicitly used in much of the "analog" literature concerning film grain [2,14]. In such a context, the model is defined as the union of a sequence of disks whose centers are randomly distributed in $\mathbb{R}^2$. The disks represent the silver halide grains in the film emulsion, and the density of the disks is chosen to respect the input image gray-level at each pixel. Thus, the model is defined in a *continuous* manner. Let us now formally present this model. Let u be an input image of size $m \times n$, with gray-levels normalized to the range $[0, 1)$, and let r be the radius of the disks of the Boolean model (in pixel "units"). Let $\mathcal{P} = \{z_i,\ i \in \mathbb{N}\} \subset [0, m] \times [0, n]$ be the Poisson process with intensity measure $\mu(dt) = \lambda(t)dt$. In the present context, the z_i's represent the centers of the film grains. The intensity λ is defined as the following piecewise

[1] Please note that the Boolean model is in fact defined in a much more general fashion, but for our purposes this definition is sufficient.

constant function

$$\lambda(x) = \frac{1}{\pi r^2} \log \frac{1}{1 - u(\lfloor x \rfloor)},$$

where $\lfloor x \rfloor$ corresponds to the pixel index p such that $x \in p + [0,1[^2$. This manner of defining λ is chosen to ensure that within each pixel domain $p + [0,1[^2$, the expected area of the Boolean model corresponds to the image gray-level $u(p)$, thus preserving the local "average" grey-level. Finally, let ϕ be some blurring filter (i.e. $\phi \geq 0$ and $\int_{\mathbb{R}^2} \phi = 1$).

Definition 1 (Inhomogeneous Boolean model associated with a digital image). *With the above notations, the* (inhomogeneous) Boolean model *associated with u is the random set Z that consists of the union of all the balls of radius r centered at the points z_i of the Poisson process $\mathcal{P}$, that is,*

$$Z = \bigcup_{i \in \mathbb{N}} \mathcal{B}_r(z_i) \subset \mathbb{R}^2.$$

Denoting by $\mathbb{1}_Z$ the indicator function of the Boolean model, that is, $\mathbb{1}_Z(x)$ equals 1 if $x \in Z$ and 0 otherwise, the filtered Boolean model *associated with u is the random field*

$$\phi * \mathbb{1}_Z(x) = \int_{\mathbb{R}^2} \mathbb{1}_Z(x-t)\phi(t)dt, \quad x \in \mathbb{R}^2.$$

We note that, while the digital input image u is discrete, both the Boolean model Z and the filtered Boolean model $\phi * \mathbb{1}_Z$ are defined in the continuous domain $\mathbb{R}^2$. Of course in practice, one is interested to produce a sampling of $\phi * \mathbb{1}_Z$ on a discrete grid to obtain a digital image. The first main objective of this work is the theoretical analysis of the grain model from a statistical point of view, in particular the first and second order statistics, the latter being a distinguishing feature of textures.

Proposition 1 (Expected Value and Covariance of the Filtered Boolean Model). *Consider a Boolean model Z with underlying Poisson process $\mathcal{P}$ having intensity measure $\mu : A \mapsto \int_A \lambda(t)dt$. Let ϕ represent a blurring filter. Then for all $x, y \in \mathbb{R}^2$,*

$$\mathbb{E}\left[\mathbb{1}_Z(x)\right] = 1 - \mathbb{E}\left[\mathbb{1}_{Z^c}(x)\right] = 1 - \exp\left(-\mu(\mathcal{B}_r(x))\right) \tag{1}$$

$$\mathrm{Cov}(\mathbb{1}_Z)(x,y) = \exp(-\mu(\mathcal{B}_r(x)) - \mu(\mathcal{B}_r(y)))\Big(\exp(\mu(\mathcal{B}_r(x) \cap \mathcal{B}_r(y))) - 1\Big). \tag{2}$$

Hence, due to the linearity of the convolution with filter ϕ, the expected value and covariance of the filtered Boolean model are given by

$$\mathbb{E}\left[\phi * \mathbb{1}_Z(x)\right] = \phi * \mathbb{E}\left[\mathbb{1}_Z\right](x) = 1 - \int_{\mathbb{R}^2} \exp(-\mu(\mathcal{B}_r(x-t)))\phi(t)dt \tag{3}$$

$$\mathrm{Cov}(\phi * \mathbb{1}_Z)(x,y) = \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} \mathrm{Cov}(\mathbb{1}_Z)(x-s, y-t)\phi(s)\phi(t)dsdt. \tag{4}$$

Proof. The second part of the proposition is straightforward, so here we give a detailed proof of Equation (1) and 2. Let us first consider the expectation. Clearly, $\mathbb{E}\left[\mathbb{1}_Z(x)\right] = 1 - \mathbb{E}\left[\mathbb{1}_{Z^c}(x)\right]$ since $\mathbb{1}_Z(x) = 1 - \mathbb{1}_{Z^c}(x)$. Note that for any point x, $\mathbb{1}_{Z^c}(x)$ is only equal to 1 if *no* balls cover x, that is,

$$\mathbb{1}_{Z^c}(x) = \prod_{z_i \in \mathcal{P}} \mathbb{1}_{\mathcal{B}_r^c(z_i)}(x).$$

Hence one can compute $\mathbb{E}\left[\mathbb{1}_{Z^c}(x)\right]$ by invoking the following general formula. In general, for any Poisson process $\mathcal{P}$ with intensity measure Θ, and for any measurable function $f : E \to [0,1]$, one has [13, page 65]

$$\mathbb{E}\left[\prod_{z_i \in \mathcal{P}} f(z_i)\right] = \exp\left(\int_{\mathbb{R}^2} (f-1) d\Theta\right) \tag{5}$$

In our case, we have $\Theta = \mu$ and $f(z) = \mathbb{1}_{\mathcal{B}_r^c(z)}(x) = \mathbb{1}_{\mathcal{B}_r^c(x)}(z)$, thus,

$$\mathbb{E}\left[\mathbb{1}_{Z^c}(x)\right] = \exp\left(\int_{\mathbb{R}^2} \left(\mathbb{1}_{\mathcal{B}_r^c(x)}(z) - 1\right) \lambda(z) dz\right) = \exp\left(-\mu(\mathcal{B}_r(x))\right),$$

which proves Equation (1). Let us now turn to the computation of the covariance. Since $\mathbb{1}_{Z^c} = 1 - \mathbb{1}_Z$ and the covariance is invariant by the multiplication by -1 and the addition of a constant, one has $\mathrm{Cov}(\mathbb{1}_Z)(x,y) = \mathrm{Cov}(\mathbb{1}_{Z^c})(x,y)$. Now,

$$\mathrm{Cov}(\mathbb{1}_{Z^c})(x,y) = \mathbb{E}\left[\mathbb{1}_{Z^c}(x)\mathbb{1}_{Z^c}(y)\right] - \mathbb{E}\left[\mathbb{1}_{Z^c}(x)\right]\mathbb{E}\left[\mathbb{1}_{Z^c}(y)\right].$$

and we need to evaluate $\mathbb{E}\left[\mathbb{1}_{Z^c}(x)\mathbb{1}_{Z^c}(y)\right]$. Using the above expression of $\mathbb{1}_{Z^c}(x)$

$$\mathbb{1}_{Z^c}(x)\mathbb{1}_{Z^c}(y) = \prod_{z_j \in \mathcal{P}} \mathbb{1}_{\mathcal{B}_r^c(x) \cap \mathcal{B}_r^c(y)}(z_j)$$

Using again Equation (5) with $f(z) = \mathbb{1}_{\mathcal{B}_r^c(x) \cap \mathcal{B}_r^c(y)}(z) = \mathbb{1}_{\mathcal{B}_r^c(x) \cap \mathcal{B}_r^c(y)}(z)$ one has

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}_{Z^c}(x)\mathbb{1}_{Z^c}(y)\right] &= \exp\left(\int_{\mathbb{R}^2} (\mathbb{1}_{\mathcal{B}_r^c(x) \cap \mathcal{B}_r^c(y)}(z) - 1)\ \lambda(z)\ dz\right) \\ &= \exp(-\mu(\mathcal{B}_r(x) \cup \mathcal{B}_r(y))).\end{aligned}$$

Hence,

$$\begin{aligned}\mathrm{Cov}(\mathbb{1}_Z)(x,y) &= \exp(-\mu(\mathcal{B}_r(x) \cup \mathcal{B}_r(y))) - \exp\left(-\mu(\mathcal{B}_r(x))\right)\exp\left(-\mu(\mathcal{B}_r(y))\right) \\ &= \exp(-\mu(\mathcal{B}_r(x)) - \mu(\mathcal{B}_r(y)))\Big(\exp(\mu(\mathcal{B}_r(x) \cap \mathcal{B}_r(y))) - 1\Big).\end{aligned}$$

□ Before continuing, let us summarize the theoretical results presented here. Firstly, we have shown that, in terms of covariance, the "positive" and "negative" Boolean grain models are equivalent, in other words, the covariance of the texture produced in dark

regions or light regions will be symmetric with respect to the "middle" gray-level. Secondly, that this covariance is dependent on the input image gray-level, which means that methods that rely on grain scanned at a given resolution are inherently incorrect. Another remark is that in the case of the *unfiltered* Boolean model, when $||x - y|| \geq 2r$, we have $\mathrm{Cov}(\mathbb{1}_Z(x), \mathbb{1}_Z(y)) = 0$. This is coherent with what we expect from the Boolean model, and will be useful further on. Finally, we have given the exact expression and an approximation method of the expected value and covariance of the filtered Boolean model.

3 Gaussian approximation of the filtered Boolean model

The second main objective of this work is to propose an approximation of the filtered Boolean model using Gaussian textures. This requires the evaluation of the expected value and the covariance of the model for all pixels on a grid. Unfortunately, the expressions given in Equation (3) and Equation (4) cannot be evaluated exactly. However, we can approximate them using a Monte Carlo integration.

3.1 Monte Carlo Integration for Approximating the Expected Value and Covariance of the Filtered Boolean Model

We will carry out two Monte Carlo integrations, one for the expected value, and one for the covariance. Let M and N be the number of samples for these Monte Carlo integrations, and $\{\xi_1 \dots \xi_M\}$ and $\{\xi'_1 \dots \xi'_N\}$ be two sequences of independently and identically distributed (i.i.d.) standard normal variables. Using the law of large numbers, we have

$$\frac{1}{M} \sum_{k=1}^{M} \exp[-\mu(\mathcal{B}_r(x - \xi_k))] \xrightarrow[N \to +\infty]{} \mathbb{E}[\phi * \mathbb{1}_Z^c(x)], \tag{6}$$

almost surely. This gives us a straightforward method to estimate $\mathbb{E}[\phi * \mathbb{1}_Z(x)]$. We now consider the approximation of the covariance function. Recall that the final goal of this is to create a covariance matrix which will be used to produce an output image with the same covariance as a filtered Boolean grain model.

Definition 2. *We define the approximate covariance function* $\mathrm{Cov}_N(x, y)$ *as the approximation of* $\mathrm{Cov}(\phi * \mathbb{1}_Z)$ *evaluated at the couple of positions* (x, y)

$$\mathrm{Cov}_N(x, y) = \frac{1}{N^2} \sum_{k,\ell=1}^{N} \mathrm{Cov}(\mathbb{1}_Z)(x - \xi'_k, y - \xi'_\ell). \tag{7}$$

Proposition 2. *The function* Cov_N *is symmetric, positive semidefinite, and* $\mathrm{Cov}_N(x, y)$ *converges almost surely towards* $\mathrm{Cov}(\phi * \mathbb{1}_Z)(x, y)$ *when* $N \to +\infty$.

Proof. The proof of symmetry is direct. For the positivity, we have to check that for every integer d, every $(\alpha_1, \ldots, \alpha_d) \in \mathbb{R}^d$ and every $(x_1, \ldots, x_d) \in (\mathbb{R}^2)^d$, $\sum_{i,j=1}^{d} \alpha_i \alpha_j \mathrm{Cov}_N(x_i, x_j) \geq 0$. Now, it is straightforward to show that for fixed values of $\xi'_1, \ldots, \xi'_N$,

$$\sum_{i,j=1}^{d} \alpha_i \alpha_j \mathrm{Cov}_N(x_i, x_j) = \mathrm{Var}\left[\sum_{i=1}^{d} \sum_{k=1}^{N} \alpha_i \mathbb{1}_Z(x_i - \xi'_k)\right] \geq 0. \tag{8}$$

As for the convergence, a direct application of the strong law of large number for u-statistics [8] shows that the part of this sum containing only couples (k, l) of distinct integers $(k \neq l)$ converges almost surely towards its expectation $\mathrm{Cov}(\phi * \mathbb{1}_Z)(x, y)$ when $N \to +\infty$. Since the part of the sum composed of couples (k, k) is bounded by $\frac{N}{N^2}$, the whole sum converges almost surely towards the desired covariance. □

3.2 Gaussian Texture Approximation for Grain on an Input Image

As previously mentioned, we propose to approximate analog film grain with a Gaussian texture, the latter being especially good at modeling "micro-textures" [5], of which film grain is a very good example. Recall that u denotes the input image defined over the image grid $\{0, \ldots, m-1\} \times \{0, \ldots, n-1\}$ and its associated filtered Boolean model $\phi * \mathbb{1}_Z$. By computing approximations of the expected value and covariance of this model on the grid, we can produce Gaussian vectors which approximate the filtered Boolean model. These Gaussian vectors will be the output images of our algorithm. In the following, we list the pixel coordinates as $\{p_i\}$ with $i \in \{0, \ldots, mn-1\}$, and $p_i \in \mathbb{R}^2$. Vectors and matrices will be denoted with bold font. The approximation of the expectation $\mathbb{E}[\phi * \mathbb{1}_Z(p_i)]$ on a pixel p_i of the image grid is denoted by $\hat{\mathbf{u}}_\mathbf{i}$ and computed thanks to the Monte Carlo integration (6)

$$\hat{\mathbf{u}}_\mathbf{i} = 1 - \frac{1}{M} \sum_{k=1}^{M} \exp[-\mu(\mathcal{B}_r(p_i - \xi_k))].$$

In order to compute this sum, we consider first of all the following vectors:

- $\boldsymbol{\lambda} \in \mathbb{R}^{mn}$ such that $\boldsymbol{\lambda}_i = \frac{1}{\pi r^2} \log(\frac{1}{1-u(p_i)})$;
- $\mathbf{1}$: a vector of ones.

Next, we define the matrix $\mathbf{A}^{p_i} \in \mathbb{R}^{M,mn}$, with $p_i \in \mathbb{R}^2$ such that

$$\mathbf{A}^{p_i}_{k,\ell} = \mathcal{A}(\mathcal{B}_r(p_i - \xi_k) \cap (p_\ell + [0, 1[^2)), \tag{9}$$

where $\mathcal{A}$ is the Lebesgue measure in $\mathbb{R}^2$. In other words, $\mathbf{A}^{p_i}_{k,\ell}$ is the area of the part of the disk $\mathcal{B}_r(p_i - \xi_k)$ which is contained in the pixel region $p_\ell + [0, 1[^2$. Using this matrix, one has $\mu(\mathcal{B}_r(p_i - \xi_k)) = \mathbf{A}^{p_i}_{k,\cdot} \boldsymbol{\lambda}$, that is, computing the intensity measure of the ball $\mathcal{B}_r(p_i - \xi_k)$ boils down to a matrix-vector multiplication.

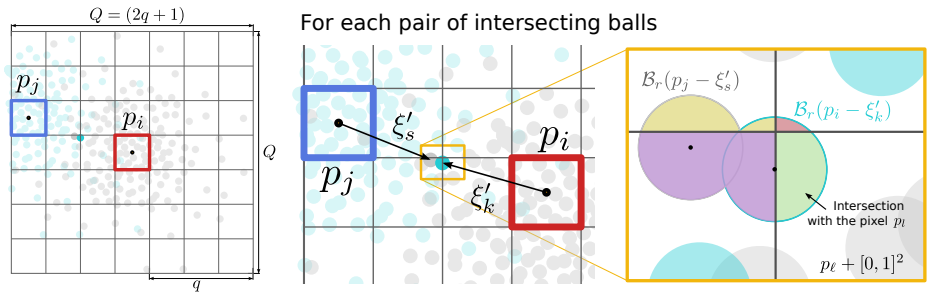

Fig. 1: **Illustration of the approximation of covariance of filtered Boolean model**. In this Figure, we illustrate the manner in which the covariance is approximated, using a Monte Carlo simulation.

Thus, the vector $\hat{\mathbf{u}}$ which approximates the expected value of the filtered Boolean model can be written

$$\hat{\mathbf{u}}_i = 1 - \frac{1}{M}\mathbf{1}^T \exp[-\mathbf{A}^{p_i}\boldsymbol{\lambda}]. \tag{10}$$

We now turn to the covariance matrix $\mathbf{C}$. The entry (i,j) of $\mathbf{C}$ is defined as the approximate covariance function evaluated at the points (p_i, p_j). In short, $\mathbf{C}_{i,j} = \mathrm{Cov}_N(p_i, p_j)$. Similarly to the case of the expectation, we define the matrices $\mathbf{B}^{p_i}$, $\mathbf{D}^{p_j}$ and $\mathbf{D}^{p_i \cap p_j}$ (this time in $\mathbb{R}^{N^2 \times mn}$), such that, for $(k,s) \in \{0,\dots,N-1\} \times \{0,\dots,N-1\}$

$$\begin{aligned}
\mathbf{B}^{p_i}_{k+Ns,\ell} &= \mathcal{A}(\mathcal{B}_r(p_i - \xi'_k) \cap (p_\ell + [0,1[^2)) \\
\mathbf{D}^{p_j}_{k+Ns,\ell} &= \mathcal{A}(\mathcal{B}_r(p_i - \xi'_s) \cap (p_\ell + [0,1[^2)) \\
\mathbf{D}^{p_i \cap p_j}_{k+Ns,\ell} &= \mathcal{A}(\mathcal{B}_r(p_i - \xi'_k) \cap \mathcal{B}_r(p_i - \xi'_s) \cap (p_\ell + [0,1[^2))
\end{aligned} \tag{11}$$

Finally, given these matrices, we can define the entry (i,j) of $\mathbf{C}$ as

$$\mathbf{C}_{i,j} = \frac{1}{N^2}\mathbf{1}^T \exp\left(-\left(\mathbf{B}^{p_i} + \mathbf{D}^{p_j}\right)\boldsymbol{\lambda}\right) \odot \left[\exp\left(\mathbf{D}^{p_i \cap p_j}\boldsymbol{\lambda}\right) - \mathbf{1}\,\right], \tag{12}$$

where $\odot$ represents the Hadamard (element-wise) vector product. Proposition 2 ensures that $\mathbf{C}$ is symmetric semi-definite positive. The covariance approximation process is illustrated in Figure 1. An interesting feature is that we can precompute these area matrices for a given parameter set, since they are independent of the image u. Furthermore, it seems reasonable to assume that the covariance will be zero for couples (p_i, p_j) which are further apart than a certain distance, by choosing a blurring kernel ϕ with compact support. In practice, we choose a truncated Gaussian for ϕ , which is truncated at the value P_α such that, for $\xi \sim N(0,1)$,

$$\mathbb{P}\big[\xi \in [-P_\alpha, P_\alpha]^2\big] = 1 - \alpha, \tag{13}$$

Algorithm 1 Film grain rendering algorithm with Gaussian texture.

Data: $u : \{0, 1, \dots m-1\} \times \{0, 1, \dots, n-1\} \to [0, u_{max}]$: input image
Parameters:
σ: standard deviation of the Gaussian low-pass filter
P_α: Gaussian $(1-\alpha)$th quantile
N: number of iterations in the Monte Carlo method
Result: Image rendered with film grain

$\mathbf{X} \sim \mathcal{N}(0, \mathbf{I}^{mn,mn})$
$q = \lfloor 2(P_\alpha + r) \rfloor$
$\xi, \xi' \leftarrow$ i.i.d. r.v. with Gaussian density truncated at P_α
$\Psi_0 \leftarrow$ computeLocalNeighbourhood(q)
Load or compute the area matrices for this parameter set
$\{\mathbf{A}^0, \mathbf{B}^0, \dots, \mathbf{D}^{\max(\Psi_0)}, \dots, \mathbf{D}^{0 \cap \max(\Psi_0)}\} \leftarrow$ AreaMatrices(ξ, ξ', r, σ)
foreach $(i, j) \in \{0, \dots, mn-1\} \times \{0, \dots, mn-1\}$ *s.t.* $\mathrm{Cov}(\phi * \mathbb{1}_Z)(p_i, p_j) \neq 0$ **do**
  $\boldsymbol{\lambda}^{p_i} \leftarrow u(\Psi_{x_i})$
  $\hat{\mathbf{u}}_i \leftarrow \frac{1}{M} \mathbf{A}^0 \boldsymbol{\lambda}^{p_i}$
  $\mathbf{C}_{p_i,p_j} = \frac{1}{N^2} \mathbf{1}^T \exp\left(-\left(\mathbf{B}^0 + \mathbf{D}^{p_j - p_i}\right) \boldsymbol{\lambda}^{p_i}\right) \odot \left[\exp\left(\mathbf{D}^{0 \cap (p_j - p_i)} \boldsymbol{\lambda}^{p_i}\right) - \mathbf{1}\right]$
$\mathbf{L} = \mathrm{Chol}(\mathbf{C})$
return$(\hat{\mathbf{u}} + \mathbf{L}\mathbf{X})$

for some small parameter α. This is the $(1 - \frac{\alpha}{2})th$ quantile of the Gaussian distribution. Now, recall that for any couple (p_i, p_j) we have

$$\mathcal{B}_r(p_i) \cap \mathcal{B}_r(p_j) = \emptyset \implies \mathrm{Cov}(\mathbb{1}_Z(p_i), \mathbb{1}_Z(p_j)) = 0. \tag{14}$$

This equation, combined with the fact that our Gaussians are truncated at P_α implies that for a couple (p_i, p_j) we have

$$||p_i - p_j||_2 > 2(P_\alpha + r) \implies \mathbf{C}_{i,j} = 0. \tag{15}$$

Let us denote with q the maximum output pixel distance for which the covariance is non-zero. This distance is

$$q = \lfloor 2(P_\alpha + r) \rfloor. \tag{16}$$

Let $Q = (2q+1)$. For any pixel p_i, the (non-zero) covariance values are therefore limited to a square neighbourhood Ψ_{p_i} of size Q^2.

Now that we have limited the extent of the covariance function, we can drastically decrease the size of the area matrices. Furthermore, these matrices only depend on the relative position of p_j with respect to p_i. Therefore, we can set p_i to be the "origin" 0 and $p_j - p_i \in \Psi_0$. In this case, we only need to calculate the matrices $\mathbf{A}^0$, $\mathbf{B}^0$, $\mathbf{D}^{p_j - p_i}$, and $\mathbf{D}^{0 \cap p_j - p_i}$. Let $\boldsymbol{\lambda}^{p_i}$ represent the values of $\boldsymbol{\lambda}$ in the neighbourhood Ψ_{p_i}. We can now rewrite the expected value and covariance

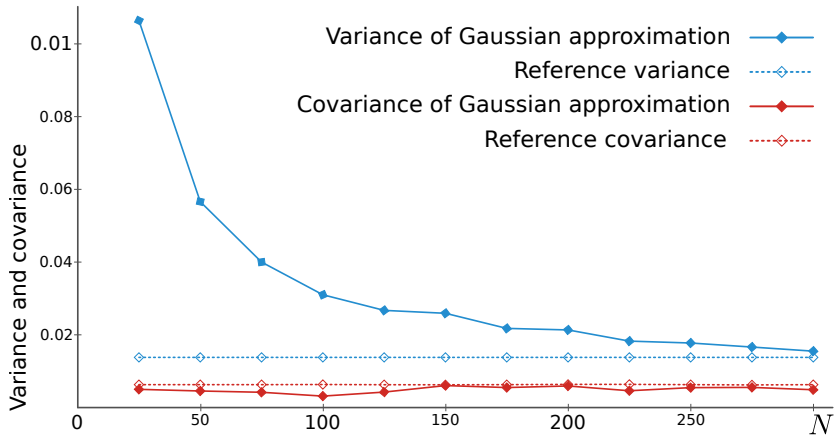

Fig. 2: **Analysis of variance and covariance of the Gaussian approximation of the Boolean model**. In this Figure, we analyze the evolution of the variance and covariance of the Gaussian approximation of the Boolean model, as N increases. As predicted, for small values of N, the approximation is biased, due to the couples (ξ'_k, ξ'_k) in the Monte Carlo simulation. This effect diminshes as N increases.

using this reduced number of vectors and matrices

$$\mathbf{C}_{i,j} = \frac{1}{N^2}\mathbf{1}^T \exp\left(-\left(\mathbf{B}^0 + \mathbf{D}^{p_j - p_i}\right)\boldsymbol{\lambda}^{p_i}\right) \odot \left[\exp\left(\mathbf{D}^{0 \cap (p_j - p_i)}\boldsymbol{\lambda}^{p_i}\right) - \mathbf{1}\right]. \tag{17}$$

This is the final expression of the covariance matrix used in our algorithm. Once the positive semi-definite covariance matrix $\mathbf{C}$ and expected value $\hat{\mathbf{u}}$ are computed, we can easily produce Gaussian vectors with these specific expected value and covariance matrix. Indeed, consider the lower triangular matrix $\mathbf{L}$ resulting from the Cholesky decomposition of $\mathbf{C}$, such that $\mathbf{C} = \mathbf{L}\mathbf{L}^T$. For any $\mathbf{X} \in \mathbb{R}^{nm}$ following a standard Gaussian white noise, the vector $\hat{\mathbf{u}} + \mathbf{L}\mathbf{X}$ has expected value $\hat{\mathbf{u}}$ and covariance matrix $\mathbf{C}$.

Algorithm summary and parameters We now recap the full film grain synthesis algorithm. This consists of two stages: firstly the computation of the area matrices. These matrices can be pre-computed (for a given parameter set), and then stored in memory. This, in turn, requires the areas of disks and intersections of disks in the ranges of the pixels of the image grid. These areas are calculated with the geometry software CGAL [15]. The second part of the full method calculates the non-zero elements of the matrix $\mathbf{C}$, carries out the Cholesky decomposition on the latter, and then produce the output image. The complete algorithm is presented in Algorithm 1. In the experiments shown in the next section, we use the following parameters: $\sigma = 0.8$, $M = 800$, $N = 200$. The radius parameter r is varied to show different grain qualities.

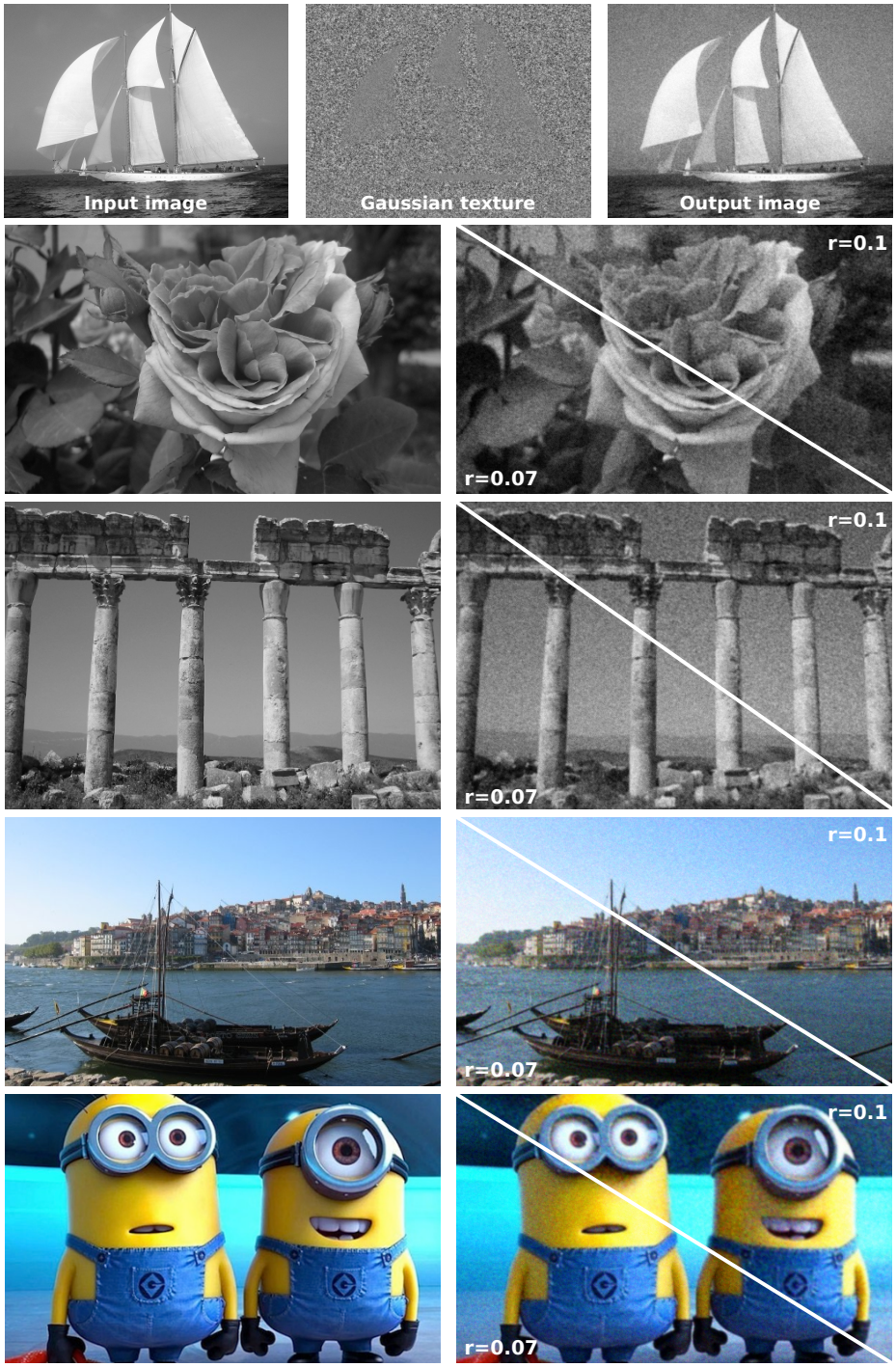

Fig. 3: **Gaussian approximation of film grain.** In this Figure, we show a result of our film grain algorithm on several input images, in gray-scale and color. In the first example of the boat, we show the "pure" texture **LX** which is added to the image. This texture has a variance which is maximized in the areas of middling gray-level (the sky), and is minimal in the areas of extreme gray-level (the boat's sails, for example).

4 Results

In this Section, we show some visual and numerical results of our algorithm. The first step is to verify experimentally that our Monte Carlo approach converges to the correct statistics of the Boolean model. One particular drawback of our approach is that, since we must use the same Gaussian offsets ξ'_i (in order to ensure symmetry and positive-definiteness), there is a list of couples (ξ'_k, ξ'_k) which are not i.i.d, but whose influence on the approximation diminishes as $N \to \infty$. With small values of N, this influence is significant, since the quantity $\mathrm{Cov}(\mathbb{1}_Z(p_i - \xi'_k), \mathbb{1}_Z(p_j - \xi'_k))$ (see Equation (7)) is maximized precisely when $i = j$, and with small N there are not enough samples to "rectify" this bias. We note, however, that this problem is mostly restricted to the case of the *variance*. Indeed, for (x, y), s.t. $||p_i - p_j||_2 > 2r$, the quantity $\mathrm{Cov}(\mathbb{1}_Z(p_i - \xi'_k), \mathbb{1}_Z(p_j - \xi'_k))$ is *necessarily* equal to 0. Only in the case of large zooms and/or large radii, will we have a non-zero influence of the couples (ξ'_k, ξ'_k). Thus, the convergence of the covariance is much faster, and indeed changes very little as N increases. This is confirmed by numerical experiments shown in Figure 2, where we analyze the evolution of the values of the variance and covariance of the Monte Carlo approach, as N increases. These values are determined on a constant image, equal to 0.5 everywhere, and we compare the values to a "reference" value determined with the result of the film grain synthesis of [11]. The covariance shown is that of two vertically adjacent pixels. This gives us an idea of how large N should be, and also serves as a strong sanity check that our approach is indeed correct. In Figure 3, we show a result of our algorithm on several input images. In order to provide some means of "objective" validation of the proposed grain, we show the result of **LX** in the middle of the top row. This represents the "pure" texture (the variance and covariance of the Gaussian approximation). This texture is coherent with what we expect, since the variance is maximal in the areas of medium gray-level, such as the sky. In areas with more extreme gray-levels, the variance is lower (the texture is smoother), which is coherent with the Boolean model. Indeed, when there are very few or very many balls in the model there is very little variation, leading to a lower value of $\mathbf{C}_{i,i}$, for any p_i in this region. This serves as a verification that the Gaussian approximation indeed displays the characteristics which we are looking for. Finally, we have shown an example of film grain on an animation image to illustrate the kind of visual style which can be achieved with our approach even on modern images.

5 Conclusion

We have presented a theoretical analysis of a physically realistic film grain model, and have derived expressions for the expected value and covariance of this model. We employ these quantities to design a film grain synthesis algorithm based on Gaussian textures. An important secondary result of this analysis is the confirmation that the covariance of film grain is dependent on the image intensity, which means that simply scanning and blending film grain is insufficient. We have

presented numerical experiments which confirm that the proposed Gaussian texture accurately imitates the second-order statistics of the film grain model, and we have shown several visual results of our approach. While the proposed algorithm produces good visual results, it is as yet limited to medium-sized images (maximum 512×512) due to memory limitations. Future work will consist in proposing a simplification of the Gaussian texture approach which exploits the statistical information presented in the current work yet has reduced computational complexity.

References

1. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. In: ACM Transactions on Graphics. vol. 25 (2006)
2. Bayer, B.E.: Relation Between Granularity and Density for a Random-Dot Model. Journal of the Optical Society of America 54(12), 1485+ (Dec 1964)
3. Chiu, S.N., Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic geometry and its applications. John Wiley & Sons, third edn. (2013)
4. DxO: Dxo film pack 5 (2016), `http://www.dxo.com/us/photography/photo-software/dxo-filmpack`
5. Galerne, B., Gousseau, Y., Morel, J.M.: Random phase textures: Theory and synthesis. IEEE Trans. Image Process. 20(1), 257 – 267 (2011)
6. G'MIC: Greyc's magic for image computing (2016), `http://gmic.eu/`
7. Grubbasoftware: Truegrain (2015), `http://grubbasoftware.com/`
8. Hoeffding, W.: The strong law of large numbers for u-statistics. Institute of Statistics mimeo series 302 (1961)
9. Livingston, R.: The Theory of the Photographic Process. By C. E. Kenneth Mees. J. Phys. Chem. 49(5), 509 (May 1945)
10. Moldovan, T.M., Roth, S., Black, M.J.: Denoising Archival Films using a Learned Bayesian Model. In: Image Processing, 2006 IEEE International Conference on. pp. 2641–2644. IEEE (Oct 2006)
11. Newson, A., Galerne, B., Delon, J.: Stochastic modelling and realistic rendering of film grain. Tech. rep., Laboratoire MAP5, Université Paris Descartes, France (2016)
12. Oh, B.T., Lei, S.M., Kuo, C.C.: Advanced Film Grain Noise Extraction and Synthesis for High-Definition Video Coding. IEEE Transactions on Circuits and Systems for Video Technology 19(12), 1717–1729 (Dec 2009)
13. Schneider, R., Weil, W.: Stochastic and Integral Geometry. Springer (2008)
14. Tanaka, K., Uchida, S.: Extended random-dot model. Journal of the Optical Society of America 73(10), 1312+ (Oct 1983)
15. The CGAL Project: CGAL User and Reference Manual. 4.9 edn. (2016), `http://doc.cgal.org/4.9/Manual/packages.html`
16. Yan, J.C.K.: Statistical methods for film grain noise removal and generation. Master's thesis, University of Toronto (1997)