



Improving Gabor Noise

Ares Lagae, Sylvain Lefebvre, Philip Dutre

► **To cite this version:**

Ares Lagae, Sylvain Lefebvre, Philip Dutre. Improving Gabor Noise. IEEE Transactions on Graphics and Visualization, IEEE, 2011, 17 (8), pp.1096-1107. <10.1109/TVCG.2010.238.>. <hal-01062522>

HAL Id: hal-01062522

<https://hal.inria.fr/hal-01062522>

Submitted on 9 Sep 2014

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.

Improving Gabor Noise

Ares Lagae, Sylvain Lefebvre, and Philip Dutré

Abstract—We have recently proposed a new procedural noise function, Gabor noise, which offers a combination of properties not found in existing noise functions. In this paper, we present three significant improvements to Gabor noise: (1) an isotropic kernel for Gabor noise, which speeds up isotropic Gabor noise with a factor of roughly two, (2) an error analysis of Gabor noise, which relates the kernel truncation radius to the relative error of the noise, and (3) spatially varying Gabor noise, which enables spatial variation of all noise parameters. These improvements make Gabor noise an even more attractive alternative for existing noise functions.

Index Terms—procedural noise, sparse convolution noise, Gabor noise, isotropic Gabor kernel, circular Gabor filter, Hankel transform, circularly symmetric functions, Gabor noise error analysis, spatially varying Gabor noise

I. INTRODUCTION

Since its introduction by Perlin in 1985, procedural noise has become an essential component in computer graphics [Ebert et al., 2002]. Several noise functions have been proposed, for example Perlin noise [Perlin, 2002], sparse convolution noise [Lewis, 1989], wavelet noise [Cook and DeRose, 2005] and anisotropic noise [Goldberg et al., 2008]. For a recent survey of procedural noise functions we refer the reader to Lagae et al.

We have recently proposed a new procedural noise function, sparse Gabor convolution noise, or, in short, *Gabor noise* [Lagae et al., 2009a] (section II). Gabor noise has several interesting properties: it is procedural, it offers significant spectral control, it supports anisotropy, it can be mapped onto surfaces without using a parametrization, it can be filtered, and it is interactive. This combination of properties is not found in existing noise functions.

In this paper, we present three significant improvements to Gabor noise. As a first improvement, we present an isotropic kernel for Gabor noise (section III). This improvement speeds up isotropic Gabor noise with a factor of roughly two. This can result in significant savings of 3D rendering time. Indeed, in the 1990’s it was informally observed that “90% of 3D rendering time is spent in shading, and 90% of that time is spent computing Perlin noise”¹. To our knowledge, the n -dimensional real and even isotropic or circularly symmetric Gabor kernel we derive is not known in literature. Zhang et al. [2002] have presented a circular Gabor filter in the context of rotation-invariant texture segmentation, but their filter is complex (that is, it has an imaginary part), and is therefore not usable in the context of Gabor noise. As a second improvement, we present an error analysis of Gabor noise (section IV). This improvement relates the kernel truncation radius, an important quality parameter of Gabor noise, to the relative error of the noise, and replaces the

ad hoc method to choose this parameter with a more principled one. As a third improvement, we present spatially varying Gabor noise (section V). This improvement enables spatial variation of all noise parameters, a property not found in existing procedural noise functions. This allows an artist to create spatially varying procedural textures. Indeed, spatial variation of noise parameters was shown to be useful in the context of non-procedural methods related to Gabor noise [van Wijk, 1991, Ware and Knight, 1995, Holten et al., 2006].

These three orthogonal improvements build upon and augment the strong theoretical foundation of Gabor noise, and make Gabor noise an even more attractive alternative for existing noise functions. Although this paper focuses on procedural noise for computer graphics, the family of methods that Gabor noise belongs to is also relevant to visualization [van Wijk, 1991].

II. GABOR NOISE

In this section, we briefly review Gabor noise [Lagae et al., 2009a]. We focus on its procedural nature and spectral control, the two properties most relevant to this paper. Since this paper addresses improvements to Gabor noise, we assume the reader is generally familiar with Gabor noise.

Anisotropic Gabor noise is a sum of randomly weighted and positioned Gabor kernels,

$$n_{K,F_0,a,\omega_0}(x,y) = \sum_i w_i g_{K,F_0,a,\omega_0}(x-x_i, y-y_i), \quad (1)$$

where the magnitude K , the frequency F_0 , the bandwidth a and the orientation ω_0 are the noise parameters, g is the Gabor kernel, $\{w_i\}$ are the random weights, and $\{(x_i, y_i)\}$ are the random positions. The Gabor kernel is the product of a radially symmetric Gaussian and a 2D cosine,

$$g_{K,F_0,a,\omega_0}(x,y) = K \exp[-\pi a^2(x^2 + y^2)] \cos[2\pi F_0(x \cos \omega_0 + y \sin \omega_0)], \quad (2)$$

where K and a control the magnitude and width of the Gaussian, and (F_0, ω_0) is the frequency of the cosine. Anisotropic Gabor noise is the convolution of sparse white noise and the Gabor kernel,

$$n_{K,F_0,a,\omega_0}(x,y) = \left[\sum_i w_i \delta_{(x_i,y_i)} * g_{K,F_0,a,\omega_0} \right](x,y), \quad (3)$$

where the random weights $\{w_i\}$ are distributed according to a random variable W with a uniform distribution on the interval $[-1, +1]$, and the random positions $\{(x_i, y_i)\}$ are distributed according to a Poisson distribution with impulse density λ . Because sparse white noise has a constant power spectrum, the power spectrum of anisotropic Gabor noise is a scaled version of the power spectrum of the Gabor kernel. The Gabor kernel in the frequency domain is a pair of Gaussians,

$$G_{K,F_0,a,\omega_0}(f_x, f_y) = \frac{K}{2a^2} \exp \left\{ -\frac{\pi}{a^2} \left[(f_x \pm F_0 \cos \omega_0)^2 + (f_y \pm F_0 \sin \omega_0)^2 \right] \right\}, \quad (4)$$

Ares Lagae is with the Katholieke Universiteit Leuven and REVES/INRIA Sophia-Antipolis.

Sylvain Lefebvre is with REVES/INRIA Sophia-Antipolis and ALICE/INRIA Nancy.

Philip Dutré is with the Katholieke Universiteit Leuven.

¹industry lore, as relayed by J.P. Lewis

where K and a control the magnitude and width of the Gaussians, and (F_0, ω_0) and its symmetrical counterpart are the locations of the Gaussians. Because the power spectrum of anisotropic Gabor noise is a scaled version of the power spectrum of the Gabor kernel, the parameters K , F_0 , a and ω_0 directly control the power spectrum of the noise.

Isotropic Gabor noise is a sum of randomly weighted, positioned and oriented Gabor kernels,

$$n_{K, F_0, a}(x, y) = \sum_i w_i g_{K, F_0, a}(\omega_{0i}, x - x_i, y - y_i), \quad (5)$$

where the magnitude K , the frequency F_0 and the bandwidth a are the noise parameters, and the random orientations $\{\omega_{0i}\}$ are distributed according to a random variable Ω with a uniform distribution on the interval $[0, 2\pi)$. Similar to anisotropic Gabor noise, the parameters K , F_0 and a directly control the power spectrum of the noise.

The procedural evaluation of Gabor noise is similar to that of Lewis' [1989] sparse convolution noise and Worley's [1996] cellular texture basis function. Gabor noise is evaluated procedurally by truncating the Gabor kernel and introducing a grid with a cell size equal to the radius of the truncated kernel. This restricts the evaluation of the noise to the grid cell containing the point of evaluation and the eight neighboring grid cells. The Gabor kernels in each cell are generated on-the-fly using a pseudo-random number generator.

Next to its procedural nature and spectral control, Gabor noise has several other interesting properties for computer graphics: it supports anisotropy, it can be mapped onto surfaces without using a parametrization, it can be filtered, and it is interactive. This is the major difference between Gabor noise and related methods in computer graphics, such as sparse convolution noise, and related methods in visualization, such as spot noise [van Wijk, 1991, Ware and Knight, 1995].

III. AN ISOTROPIC KERNEL FOR GABOR NOISE

Isotropic Gabor noise is defined using an anisotropic Gabor kernel (see equation 5). In this section, we show that isotropic Gabor noise can also be defined using an isotropic Gabor kernel, and that the isotropic kernel has several advantages over the anisotropic kernel. Most importantly, we show that isotropic noise using the isotropic kernel is roughly two times faster than isotropic noise using the anisotropic kernel.

We assume that the reader is familiar with circularly symmetric functions (see appendix I), more specifically, with hyperspherical coordinates (see appendix I-A), the integration (see appendix I-B) and convolution (see appendix I-C) of circularly symmetric functions, and the Hankel transform (see appendix I-D).

A. The Isotropic Gabor Kernel

We define the n -dimensional isotropic Gabor kernel similar in spirit as other kinds of Gabor kernels: using a Gaussian and a harmonic, which are related by multiplication in the spatial domain, and by convolution in the frequency domain. We use the Hankel transform (see appendix I-D), which is the method of choice for working with Fourier transforms of isotropic or circularly symmetric functions. We denote the n -dimensional isotropic Gabor kernel, Gaussian, and harmonic as ${}^n_I g(r)$, ${}^n_I G_G(r)$, and ${}^n_I g_H(r)$ in the spatial domain, and as ${}^n_I G(f_r)$, ${}^n_I G_G(f_r)$, and

${}^n_I G_H(f_r)$ in the frequency domain. We summarize their relations as

$$\begin{array}{ccc} \text{spatial domain} & & \text{frequency domain} \\ {}^n_I g_G(r) & \xleftrightarrow{{}^n\mathcal{H}} & {}^n_I G_G(f_r) \\ {}^n_I g_H(r) & \xleftrightarrow{{}^n\mathcal{H}} & {}^n_I G_H(f_r) \\ \hline {}^n_I g(r) = {}^n_I g_G(r) {}^n_I g_H(r) & \xleftrightarrow{{}^n\mathcal{H}} & [{}^n_I G_G * {}^n_I G_H](f_r) = {}^n_I G(f_r) \end{array}, \quad (6)$$

where $\xleftrightarrow{{}^n\mathcal{H}}$ denotes an order- n Hankel transform pair.

The Gaussian is a Gaussian in both the spatial domain and the frequency domain. We define the Gaussian in the spatial domain, ${}^n_I g_G(r)$, as the circularly symmetric Gaussian,

$${}^n_I g_G(r) = K e^{-\pi a^2 r^2}, \quad (7)$$

where K and a are the magnitude and width of the Gaussian. The Gaussian in the spatial domain is illustrated in figure 1(a) and in figure 2(a). We obtain the Gaussian in the frequency domain, ${}^n_I G_G(f_r)$, as the order- n Hankel transform of ${}^n_I g_G(r)$,

$$\begin{aligned} {}^n_I G_G(f_r) &= {}^n\mathcal{H}[{}^n_I g_G(r)] \\ &= \frac{2\pi}{f_r^{\frac{1}{2}n-1}} \int_0^\infty K e^{-\pi a^2 r^2} J_{\frac{1}{2}n-1}(2\pi f_r r) r^{\frac{1}{2}n} dr \\ &= \frac{K}{a^n} e^{-\frac{\pi}{a^2} f_r^2}, \end{aligned} \quad (8)$$

where J_n is the order- n Bessel function of the first kind. The Gaussian in the frequency domain is illustrated in figure 1(b) and in figure 2(b).

The harmonic is typically defined as an impulse in the frequency domain, located at the principal frequency, F_0 , of the Gabor kernel. We therefore define the harmonic in the frequency domain, ${}^n_I G_H(f_r)$, as a circularly symmetric impulse,

$${}^n_I G_H(f_r) = \delta(f_r - F_0), \quad (9)$$

where F_0 is the frequency of the harmonic. The harmonic in the frequency domain is illustrated in figure 1(d) and in figure 2(d). We obtain the harmonic in the spatial domain, ${}^n_I g_H(r)$, as the order- n Hankel transform of ${}^n_I G_H(f_r)$,

$$\begin{aligned} {}^n_I g_H(r) &= {}^n\mathcal{H}[{}^n_I G_H(f_r)] \\ &= \frac{2\pi}{r^{\frac{1}{2}n-1}} \int_0^\infty \delta(f_r - F_0) J_{\frac{1}{2}n-1}(2\pi r f_r) f_r^{\frac{1}{2}n} df_r \\ &= \frac{2\pi}{r^{\frac{1}{2}n-1}} J_{\frac{1}{2}n-1}(2\pi F_0 r) F_0^{\frac{1}{2}n}. \end{aligned} \quad (10)$$

The harmonic in the spatial domain is illustrated in figure 1(c) and in figure 2(c).

We obtain the n -dimensional isotropic Gabor kernel in the spatial domain, ${}^n_I g(r)$, as the multiplication of ${}^n_I g_G(r)$ and ${}^n_I g_H(r)$,

$$\begin{aligned} {}^n_I g(r) &= {}^n_I g_G(r) {}^n_I g_H(r) \\ &= K e^{-\pi a^2 r^2} \frac{2\pi F_0^{\frac{1}{2}n}}{r^{\frac{1}{2}n-1}} J_{\frac{1}{2}n-1}(2\pi F_0 r). \end{aligned} \quad (11)$$

The kernel in the spatial domain is illustrated in figure 1(e) and in figure 2(e). We obtain the n -dimensional isotropic Gabor kernel in the frequency domain, ${}^n_I G(f_r)$, as the convolution of ${}^n_I G_G(f_r)$ and ${}^n_I G_H(f_r)$. We simplify the convolution of isotropic or circularly symmetric functions by exploiting their symmetry (see appendix I-C). First, we convolve ${}^n_I G_G(f_r)$ and ${}^n_I G_H(f_r)$ using this simplification,

$$\begin{aligned} {}^n_I G(f_r) &= [{}^n_I G_G * {}^n_I G_H](f_r) \\ &= \frac{2\pi}{\Gamma(\frac{n-1}{2})} \int_{f'_r=0}^\infty \int_{f_\phi=0}^\pi \delta(f'_r - F_0) \frac{K}{a^n} e^{-\frac{\pi}{a^2}(f_r^2 + f'^2 - 2f_r f'_r \cos f_\phi)} \\ &\quad f_r'^{n-1} \sin^{n-2} f_\phi df'_r df_\phi, \end{aligned} \quad (12)$$

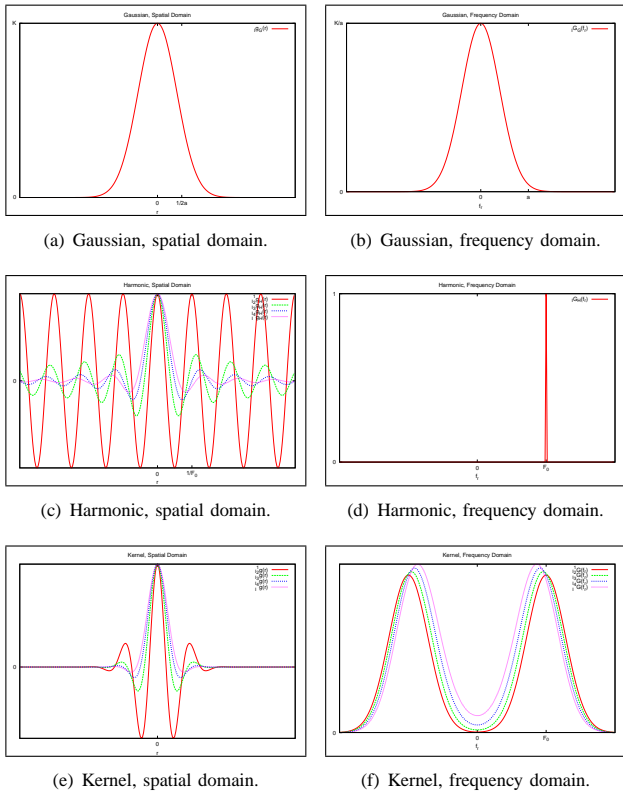


Fig. 1. The isotropic Gabor kernel. (a) Gaussian, spatial domain. (b) Gaussian, frequency domain. (c) Harmonic, spatial domain. (d) Harmonic, frequency domain. (e) Kernel, spatial domain. (f) Kernel, frequency domain.

where Γ is the Gamma function. Then, we integrate over f_r' . Next, we simplify the convolution using the integral

$$\int_0^\pi \sin^n \theta e^{x \cos \theta} d\theta = 2^{\frac{n}{2}} \sqrt{\pi} x^{-\frac{n}{2}} I_{\frac{n}{2}}(x) \Gamma\left(\frac{n+1}{2}\right), \quad (13)$$

where I_n is the order- n modified Bessel function of the first kind. Finally, we obtain

$${}_I^n G(f_r) = \frac{2\pi K F_0^{\frac{1}{2}n}}{a^2 f_r^{\frac{1}{2}n-1}} e^{\frac{-\pi}{a^2}(f_r^2 + F_0^2)} I_{\frac{1}{2}n-1}\left(\frac{2\pi F_0}{a^2} f_r\right). \quad (14)$$

The kernel in the frequency domain is illustrated in figure 1(f) and in figure 2(f).

B. Isotropic Gabor Noise using the Isotropic Gabor Kernel

We define n -dimensional isotropic Gabor noise using the isotropic Gabor kernel similar to anisotropic (see equation 1) and isotropic (see equation 5) Gabor noise using the anisotropic Gabor kernel,

$${}_I^n g(x_1, \dots, x_n) = \sum_i w_i {}_I^n g\left(\sqrt{(x_1 - x_{i,1})^2 + \dots + (x_n - x_{i,n})^2}\right) \quad (15)$$

Note that, in contrast to the anisotropic kernel, the isotropic kernel does not need to be randomly oriented. The variance of the noise ${}_I^n \sigma_n^2$ is

$${}_I^n \sigma_n^2 = \lambda E[W^2] S_n \int_{r=0}^{\infty} {}_I^n g^2(r) r^{n-1} dr. \quad (16)$$

The power spectrum of the noise ${}_I^n S_{nn}$ is

$${}_I^n S_{nn}(f_r) = \lambda E[W^2] |{}_I^n G(f_r)|^2. \quad (17)$$

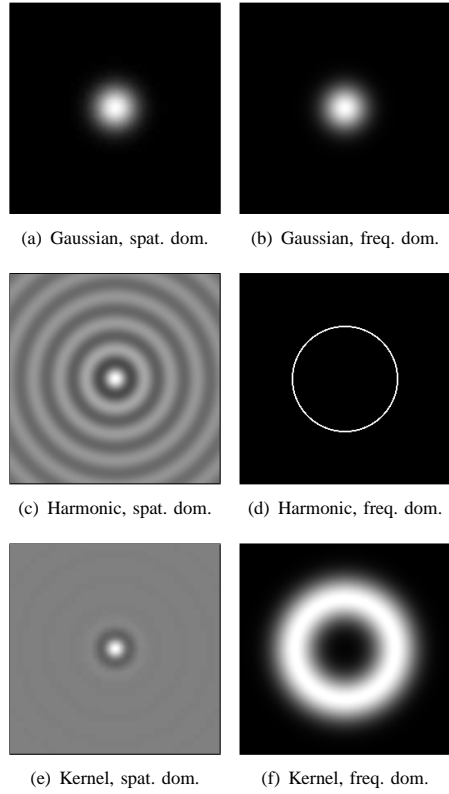


Fig. 2. The 2D isotropic Gabor kernel. (a) Gaussian, spatial domain. (b) Gaussian, frequency domain. (c) Harmonic, spatial domain. (d) Harmonic, frequency domain. (e) Kernel, spatial domain. (f) Kernel, frequency domain.

We provide equations for working with one-, two-, three- and four-dimensional isotropic Gabor noise using the isotropic Gabor kernel (see appendix II), including the isotropic Gabor kernel in the spatial domain (see appendix II-A) and in the frequency domain (see appendix II-B), the integral of the isotropic Gabor kernel squared (see appendix II-C), the envelope of the isotropic Gabor kernel (see appendix II-D), and the radius of the truncated isotropic Gabor kernel (see appendix II-E).

C. Implementation, Results and Comparison

We have implemented isotropic Gabor noise using the isotropic Gabor kernel, and we have verified most equations experimentally. We evaluate the Bessel functions using code based on Press et al. [2002, 6.5, 6.6] (polynomial approximations are also available in Abramowitz and Stegun [1972, 9.4, 9.8]), and the Lambert-W function (see appendix II-E) using code based on Keith [2009]. We verify the noise by comparing the estimated and expected power spectrum and the actual and expected intensity distribution [Lagae et al., 2009b].

We illustrate one-, two- and three-dimensional isotropic Gabor noise using the isotropic Gabor kernel in figure 3, figure 4 and figure 5. Note the similarity between isotropic noise using the isotropic kernel and isotropic noise using the anisotropic kernel [Lagae et al., 2009a, figure 4]. Also note how closely the estimated and expected power spectrum and the actual and expected intensity distribution match.

We have found that isotropic noise using the isotropic kernel is significantly faster than isotropic noise using the anisotropic kernel. This is because of two reasons. First, the isotropic kernel

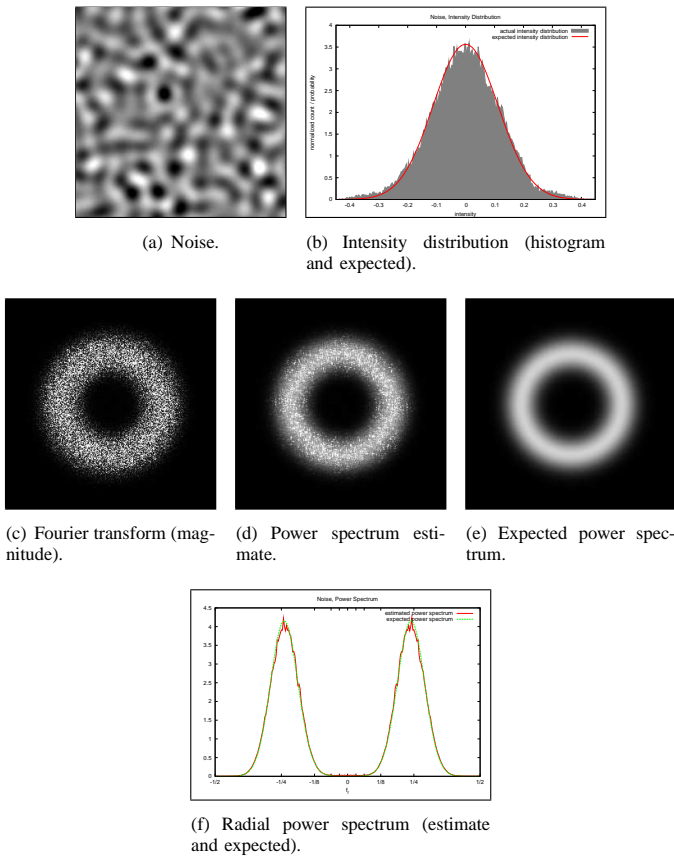


Fig. 4. 2D isotropic Gabor noise using the isotropic kernel. (a) Noise. (b) Actual and expected intensity distribution. (c) Fourier transform (magnitude). (d) Estimated power spectrum. (e) Expected power spectrum. (f) Estimated and expected radial power spectrum.

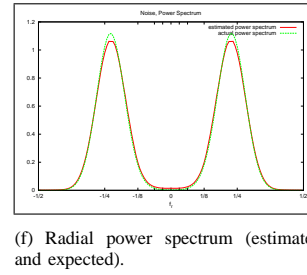
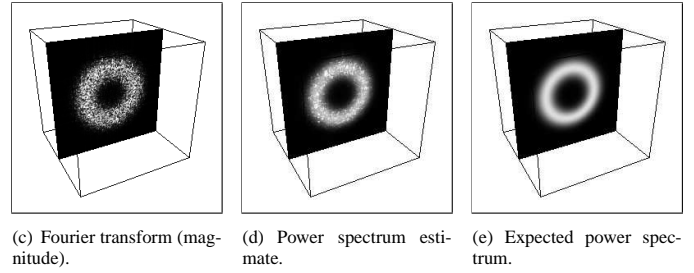
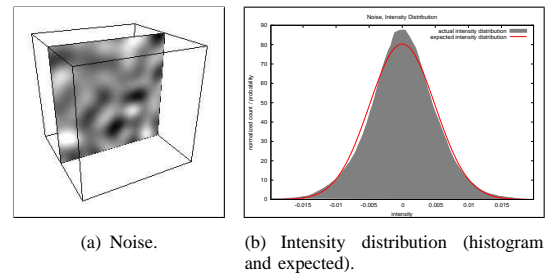


Fig. 5. 3D isotropic Gabor noise using the isotropic kernel. (a) Noise. (b) Actual and expected intensity distribution. (c) Fourier transform (magnitude). (d) Estimated power spectrum. (e) Expected power spectrum. (f) Estimated and expected radial power spectrum.

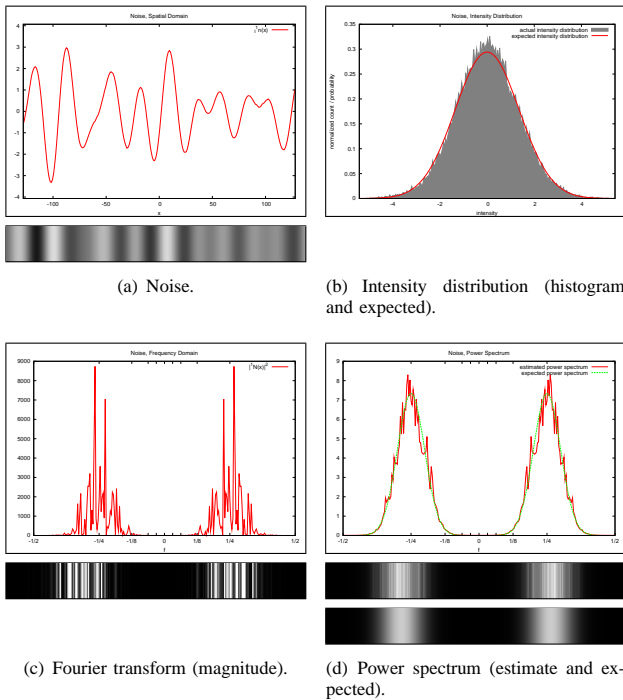


Fig. 3. 1D isotropic Gabor noise using the isotropic kernel. (a) Noise. (b) Actual and expected intensity distribution. (c) Fourier transform (magnitude). (d) Estimated and expected power spectrum.

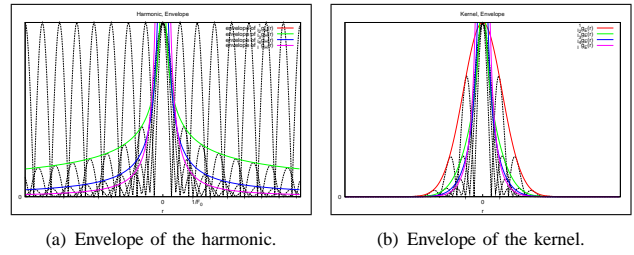


Fig. 6. The envelope of the isotropic Gabor kernel. (a) Envelope of the harmonic. (b) Envelope of the kernel.

is more compact than the anisotropic kernel (see figure 1(e) and figure 6). This results in a lower number of impulses per kernel for the same impulse density, which in turn results in a shorter time to evaluate the noise, since this time is directly proportional to the number of impulses. Second, in contrast to the anisotropic kernel, the isotropic kernel does not need to be randomly oriented (see equation 5 and equation 15). This avoids the generation of random orientations, which also results in a shorter time to evaluate the noise. For example, for two-dimensional isotropic noise with parameters $K = 0.709645$, $a = 0.0443528$ and $F_0 = 0.0625$, where the kernel was truncated at 5% of its maximum value, the radius of the kernel is 22.0169 for the anisotropic kernel, but only 15.6741 for the isotropic kernel, a reduction of 28.8086%, and for an impulse density of $\lambda = 0.0414605$, the number of impulses per kernel is 63.1386 for the anisotropic kernel, but only 32 for the

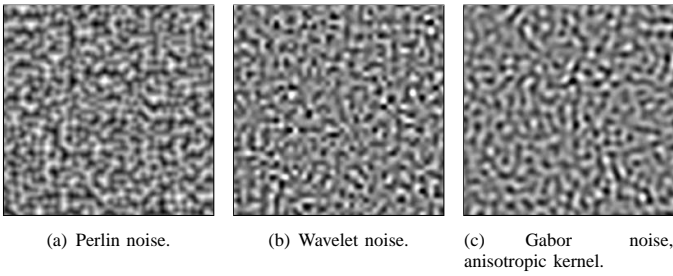


Fig. 7. A comparison of visual quality of different noise functions. (a) Perlin noise. (b) Wavelet noise. (c) Gabor noise using the isotropic kernel. Note that Perlin noise, and to a lesser degree also wavelet noise, both exhibit an undesired axis-aligned anisotropy.

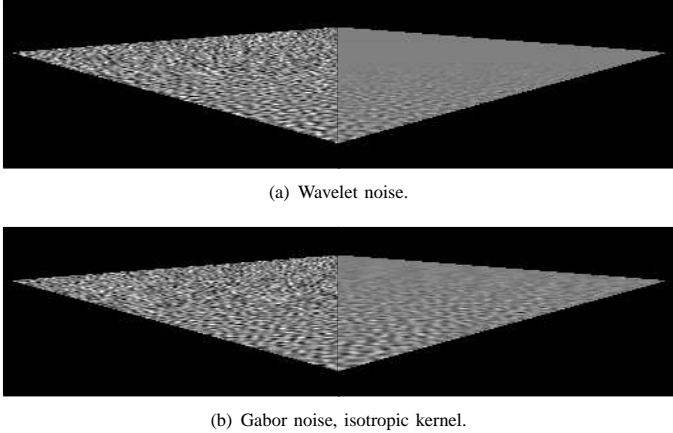


Fig. 8. A comparison of filtering quality of different noise functions. (a) Wavelet noise. (b) Gabor noise using the isotropic kernel. Each subfigure shows a tilted plane with unfiltered noise on the left half and filtered noise on the right half. Note that Gabor noise is better than wavelet noise at preserving detail at high spatial frequencies in the far distance. Please also refer to video 1.

isotropic kernel, a reduction of 49.3179%. The time to evaluate the noise using our CPU implementation is 4.11625 s for the anisotropic kernel, but only 1.64481 s for the isotropic kernel, a speedup of 2.50257 (512×512 , Intel(R) Xeon(R) CPU 5160 at 3.00GHz). With similar parameter settings, the performance of the noise using our GPU implementation is 77 FPS (frames per second) for the anisotropic kernel, and 144 FPS for the isotropic kernel, a speedup of 1.87 (512×512 , NVIDIA Quadro FX 5800 GPU). We have found that the speedup is roughly independent of the noise parameters. The speedup for three- and four-dimensional isotropic noise should be even larger, since the isotropic kernel gets more compact with increasing dimension (see figure 1(e) and figure 6), and the number of random orientations that has to be sampled gets larger with increasing dimension (see appendix I-A).

We have compared isotropic Gabor noise using the isotropic kernel with Perlin noise [Perlin, 2002] and wavelet noise [Cook and DeRose, 2005], two other isotropic noise functions. The visual quality of these noise functions is illustrated in figure 7. Perlin noise, and to a lesser degree also wavelet noise, both exhibit an undesired axis-aligned anisotropy, which is not the case for Gabor noise (also see [Lagae et al.]). We have measured the performance of these noise functions using our GPU implementations. The performance of Perlin noise is 3388 FPS, and that of wavelet noise is 730 FPS (512×512 , NVIDIA Quadro FX 5800 GPU). The performance of Gabor noise is dependent

on the parameter settings and the approximations used in the implementation. With similar parameter settings as above, the performance of Gabor noise is 144 FPS, and 255 FPS for half the number of impulses per kernel. When approximating the Poisson distribution by its mean, the performance is 198 FPS and 357 FPS. The filtering quality of wavelet noise and Gabor noise is illustrated in figure 8 and in video 1. We filter wavelet noise by deriving a filtering weight by evaluating the filtering Gaussian in the frequency domain at the principal frequency of the noise. We filter Gabor noise by deriving new kernel parameters K , F_0 and a by multiplying the filtering Gaussian in the frequency domain with a Gaussian approximation of the isotropic kernel in the frequency domain. In both cases, the anisotropic filtering Gaussian in the frequency domain is approximated with an isotropic one. Gabor noise enables a better filtering quality than wavelet noise since Gabor noise is better than wavelet noise at preserving detail at high spatial frequencies in the far distance. This is clearly visible in video 1. This is because wavelet noise only allows to apply a single filtering weight w to an entire noise octave, while Gabor noise allows to adapt the frequency f and bandwidth a of individual kernels. We have not included Perlin noise in this comparison, since Cook and DeRose [2005, figure 1] already showed that it is worse than wavelet noise in terms of filtering quality, and because filtering Perlin noise in a principled way is difficult since Perlin noise is not band-pass. We would like to note that there are many other criteria that could also be taken into account in a comparison [Lagae et al.], which might or might not be relevant depending on the application.

Our results and comparison show that isotropic Gabor noise using the isotropic Gabor kernel should be used whenever an isotropic noise function is required with a higher quality than Perlin noise and wavelet noise and a lower cost than isotropic Gabor noise using the anisotropic Gabor kernel.

IV. AN ERROR ANALYSIS OF GABOR NOISE

The procedural evaluation of Gabor noise requires that the Gabor kernel is truncated. This is typically done at a radius where the envelope of the kernel reaches a sufficiently small value, for example 5% of its maximum value [Lagae et al., 2009a, 4] [Lagae et al., 2009b, 1]. However, this is an ad hoc approach, since truncating the kernel introduces an error in the noise, and this error is not quantified. In this section, we quantify the effect of truncating the Gabor kernel, and we present a more principled approach to truncate the Gabor kernel.

A. Relation of Kernel Truncation Radius to Noise Error

We relate the noise error resulting from truncating the Gabor kernel g to the truncation radius r_t . In this analysis, we use the n -dimensional isotropic Gabor kernel (see section III), but the analysis is also valid for other kernels, such as the anisotropic Gabor kernel. We define a truncated Gabor kernel g_t , where

$$g_t(r) = \begin{cases} g(r) & 0 \leq r_t < r \\ 0 & r_t \geq r \end{cases}, \quad (18)$$

and an error kernel Δg , where

$$\Delta g(r) = \begin{cases} 0 & 0 \leq r_t < r \\ g(r) & r_t \geq r \end{cases}. \quad (19)$$

Note that $g(r) = g_t(r) + \Delta g(r)$. We obtain the noise error Δn by subtracting the noise using the truncated kernel n_t from the

noise using the untruncated kernel n ,

$$\begin{aligned} \Delta n(x_1, \dots, x_n) &= n(x_1, \dots, x_n) - n_t(x_1, \dots, x_n) \\ &= \sum_i w_i \Delta g \left(\sqrt{(x_1 - x_{1i})^2 + \dots + (x_n - x_{ni})^2} \right). \end{aligned} \quad (20)$$

Our key insight is that the noise error Δn is, similar to Gabor noise, a random pulse process [Lagae et al., 2009a, 2.2] [van Etten, 2005, 8] [Papoulis and Pillai, 2002, 10.2]. The variance $\sigma_{\Delta n}^2$ of the noise error Δn is therefore

$$\sigma_{\Delta n}^2 = \lambda E[W^2] S_n \int_0^\infty \Delta g^2(r) dr. \quad (21)$$

We define the root mean square error and the relative error using the variances σ_n^2 and $\sigma_{\Delta n}^2$ of the noise and the noise error. We define the root mean square error e_{RMS} as the square root of the variance of the noise error,

$$e_{RMS} = \sqrt{\sigma_{\Delta n}^2}. \quad (22)$$

We define the relative error e as the root mean square error e_{RMS} over the root mean square amplitude $\sqrt{\sigma_n^2}$,

$$e = \frac{e_{RMS}}{\sqrt{\sigma_n^2}} = \sqrt{\frac{\int_0^\infty \Delta g^2(r) dr}{\int_0^\infty g^2(r) dr}} = \sqrt{1 - \frac{\int_0^{r_t} g^2(r) dr}{\int_0^\infty g^2(r) dr}}. \quad (23)$$

Note that the relative error e only depends on the kernel g and the truncation radius r_t , and not on the parameters of the sparse white noise λ and W . Our analysis allows to determine the relative noise error for a given kernel truncation radius, but also allows to determine the kernel truncation radius for a given relative error, by solving equation 23 for r_t . The usage of relative error in this context is motivated by Weber's Law [Blackwell, 1972]. This is a much more principled approach than truncating the kernel at an arbitrarily chosen value.

B. Implementation, Results and Discussion

We have implemented the noise error analysis and we have verified the equations experimentally. For most kernels, a closed-form expression for $\int_0^\infty g^2(r) dr$ is available (see appendix II-C), but a closed-form expression for $\int_0^{r_t} g^2(r) dr$ is not. Therefore, we generally solve equation 23 for r_t numerically, using bracketing and bisection [Press et al., 2002, 9.1], the closed-form expression to evaluate $\int_0^\infty g^2(r) dr$, and Simpson's rule [Press et al., 2002, 4.2] to evaluate $\int_0^{r_t} g^2(r) dr$. Note that for isotropic noise using the isotropic kernel, all integrals are one-dimensional.

We illustrate the error analysis of one-dimensional isotropic noise using the isotropic kernel in figure 9. Note how closely the actual and expected root mean square error match. We illustrate the error analysis of two-dimensional isotropic noise using the isotropic kernel in figure 10. We plot the relative error versus the kernel radius for one-, two-, three- and four-dimensional isotropic noise using the isotropic kernel in figure 11. Note that the relative error quickly decreases with increasing kernel truncation radius.

We now revisit the example of subsection III-C. When truncating the kernel using a relative error of 2%, the radius of the kernel is 25.25 for the anisotropic kernel, but only 20.8984 for the isotropic kernel, a reduction of 17.2341%, the number of impulses per kernel is 46.7139 for the anisotropic kernel, but only 32 for the isotropic kernel, a reduction of 31.498%, and the time to evaluate the noise using our CPU implementation is 2.50428 s for the anisotropic kernel, but only 1.02537 s for the isotropic kernel, a speedup of 2.44233. We generalize this example and

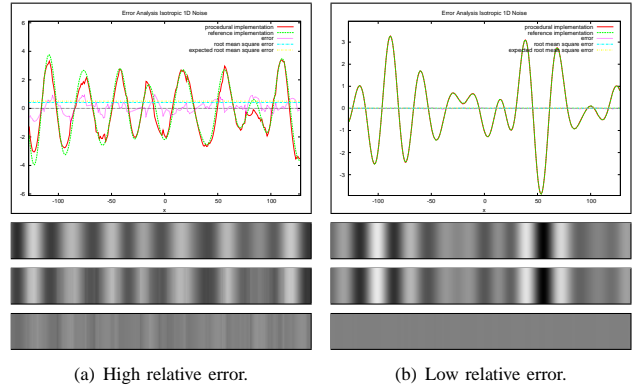


Fig. 9. An error analysis of 1D isotropic Gabor noise. (a) Noise with a high relative error ($e = 25\%$). (b) Noise with a low relative error ($e = 1\%$). Each subfigure shows a graph and three images. The graph shows the noise using the untruncated kernel (n) (reference implementation), the noise using the truncated kernel (n_t) (procedural implementation), the noise error (Δn), the actual root mean square noise error, and the estimated root mean square noise error (e_{RMS}). The three images show the noise using the untruncated kernel (n), the noise using the truncated kernel (n_t), and the noise error (Δn).

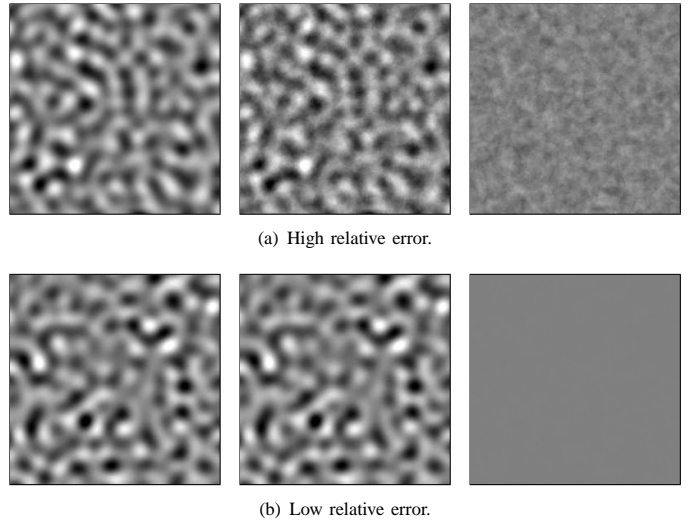


Fig. 10. An error analysis of 2D isotropic Gabor noise. (a) Noise with a high relative error ($e = 25\%$). (b) Noise with a low relative error ($e = 1\%$). Each subfigure shows the noise using the untruncated kernel (n) (reference implementation), the noise using the truncated kernel (n_t) (procedural implementation), and the noise error (Δn).

plot the relative error versus the kernel radius for two-dimensional isotropic noise using the anisotropic and the isotropic kernel in figure 12. This figure shows that for the same truncated kernel radius, the relative error is always smaller for the isotropic kernel than for the anisotropic kernel, and for the same relative error, the isotropic kernel is always smaller and therefore faster than the anisotropic kernel.

We believe that in general equation 23 should be used rather than the ad hoc approach to determine the radius of the truncated kernel, except for reasons of computational expense, for example in the case of spatially varying noise (see section V), or simplicity.

V. SPATIALLY VARYING GABOR NOISE

The procedural evaluation of Gabor noise introduces a grid with a cell size equal to the radius of the truncated kernel [Lagae et al., 2009a, 4]. This restricts the evaluation of the noise to the grid cell

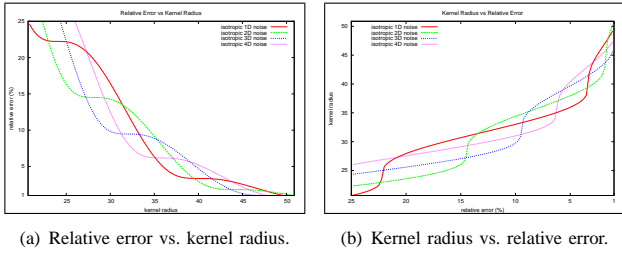


Fig. 11. Relative error versus kernel radius for isotropic Gabor noise using the isotropic kernel. (a) Relative error versus kernel radius. (b) Kernel radius versus relative error.

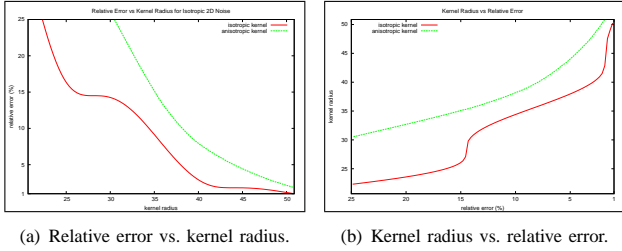


Fig. 12. Relative error versus kernel radius for 2D isotropic Gabor noise using the anisotropic and the isotropic kernel (a) Relative error versus kernel radius. (b) Kernel radius versus relative error.

containing the point of evaluation and the eight neighboring grid cells. However, this also prohibits spatial variation of the noise parameters, since the kernel radius, and therefore the cell size, is in general dependent on these parameters. In this section, we present a procedural evaluation of Gabor noise that enables noise with spatially varying parameters.

A. Procedural Evaluation of Gabor Noise with Spatially Varying Parameters

Evaluating Gabor noise using a single grid is optimal for a single kernel radius corresponding to a fixed set of noise parameters. In order to handle arbitrary kernel radii, corresponding to arbitrary or noise parameters, we use a hierarchy of grids, where the cell size of consecutive grids differs by a factor of two. More specifically, the grid at level l of the hierarchy has a cell size of 2^l . This implies that the grid at level l is optimal for a kernel radius $r_l = 2^l$. We have observed that, for different noise parameters, an equal noise quality is obtained by maintaining a constant number of impulses N per kernel area. Therefore, we associate the grid at level l with an impulse density of $\lambda_l = N/\pi r_l^2$. When evaluating the noise using an arbitrary kernel radius r , we use the two grids with kernel radii r_0 and r_1 that bracket r . These are the grids with level $l_0 = \lfloor \log_2 r \rfloor$ and $l_1 = l_0 + 1$, for which $r_0 \leq r < r_1$. We parametrize the radius r in terms of r_0 and r_1 by introducing a parameter α in $[0, 1)$ such that $r = (1 - \alpha)r_0 + \alpha r_1$. When evaluating the noise, we combine the impulses of both grids by weighing the contribution due to each grid,

$$n(x, y) = w_0(\alpha) \sum_{i_0} w_{i_0} g(x - x_{i_0}, y - y_{i_0}) + w_1(\alpha) \sum_{i_1} w_{i_1} g(x - x_{i_1}, y - y_{i_1}), \quad (24)$$

where the first and second term correspond to the grid with level l_0 and l_1 , and w_0 and w_1 are weighting functions. This is illustrated in figure 13. It is important to note that this is a linear combination of two noises with the same parameters, and that the goal of the interpolation is to transition between grids

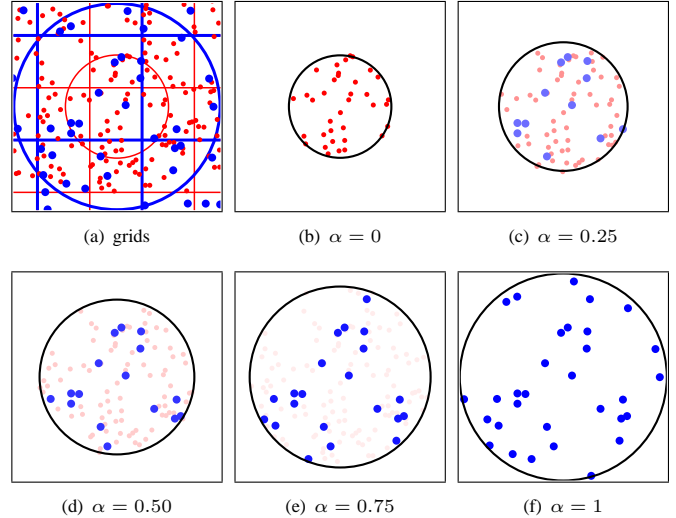


Fig. 13. The hierarchy of grids used in spatially varying Gabor noise. (a) The grids at level l_0 (red) and l_1 (blue). (b-f) The weighted combined impulses of both grids for different kernel radii. Note that the red points disappear faster than the blue points appear.

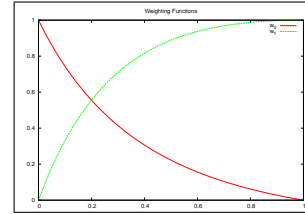


Fig. 14. The weighting functions used in spatially varying Gabor noise.

with different impulse densities, and not to obtain a noise with interpolated parameters. Interpolating between two noises with frequencies F_0 and F_1 (for example two octaves of Perlin noise) cannot produce a noise with an intermediate frequency F , due to the linearity of the Fourier transform, while interpolating between two noises that already have frequency F obviously can.

For obvious reasons, we require that $w_0(0) = 1$, that $w_0(1) = 0$, and that w_0 is monotonically decreasing, and similarly that $w_1(0) = 0$, that $w_1(1) = 1$, and that w_1 is monotonically increasing. However, several choices for w_0 and w_1 remain. A simple option is to choose $w_0(\alpha) = 1 - w_1(\alpha)$ and $w_1(\alpha) = w_1(\alpha)$, where $w_1(\alpha)$ is the linear weighting function $w_1(\alpha) = \alpha$ or the cubic weighting function $w_1(\alpha) = 3\alpha^2 - 2\alpha^3$. However, the resulting weighting functions do not result in visually pleasing transitions. This is illustrated in video 2. The weighting functions can be determined by imposing additional constraints. For example, imposing the constraint that the power spectrum of both sides of equation 24 is the same leads to $w_0(\alpha)(1 + \alpha) + w_1(\alpha)(1 + \alpha) = 1$. We have noticed that the visually unpleasing transitions are caused by the fact that the two grids have a different impulse density. The number of impulses per kernel area N_0 due to the grid with level l_0 equals $N_0 = (1 + \alpha)^2 N$, while the number of impulses per kernel area N_1 due to the grid with level l_1 equals $N_1 = N_0/4$. Therefore, we impose the constraint that the weighted number of impulses per kernel area remains constant, more specifically such that $w_0(\alpha)N_0 + w_1(\alpha)N_1 = N$ or equivalently such that $w_0(\alpha)(1 + \alpha)^2 + w_1(\alpha)(1 + \alpha)^2/4 = 1$. We now choose $w_0(\alpha)(1 + \alpha)^2 = 1 - w_1(\alpha)$ and $w_1(\alpha)(1 + \alpha)^2/4 =$

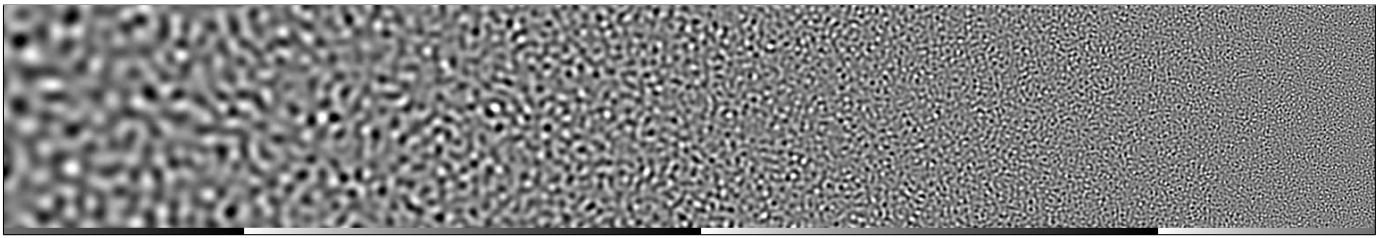


Fig. 15. Spatially varying 2D Gabor noise. The noise parameters vary from $K = 0.7096$, $F_0 = 0.03125$ and $a = 0.02218$ at the left to $K = 0.7096$, $F_0 = 0.25$ and $a = 0.1774$ at the right. The value of α is visualized at the bottom of the image. Note that this image spans four grid levels.

$w(\alpha)$, where $w(\alpha)$ is a weighting function. The simplest option for $w(\alpha)$ is the linear weighting function $w(\alpha) = \alpha$, which also satisfies the power spectrum constraint, and which results in the weighting functions

$$w_0(\alpha) = \frac{1 - \alpha}{(1 + \alpha)^2} \quad (25)$$

$$w_1(\alpha) = \frac{4\alpha}{(1 + \alpha)^2}. \quad (26)$$

This is illustrated in figure 14. These weighting functions result in visually pleasing transitions. This is illustrated in video 2.

B. Implementation, Results, Comparison and Discussion

We have implemented spatially varying Gabor noise and we have verified the equations experimentally. Note that in contrast with regular Gabor noise, the evaluation of spatially varying Gabor noise includes the computation of the kernel truncation radius. Because of computational expense, we use the ad hoc approach (see section IV). Also note that the noise evaluation is not restricted to exactly nine grid cells, and that the seeding strategy has to take into account the grid level.

We illustrate spatially varying 2D Gabor noise in figure 15 and video 2. We show procedural textures generated with spatially varying surface Gabor noise in figure 16. Note that the simple alternative of warping the domain variables of a regular Gabor noise to obtain spatially varying Gabor noise might work in the case of a 2D noise, as in figure 15, but would probably not work in the case of spatially varying surface noise, as in figure 16, and might be problematic for filtering.

We have compared the performance of spatially varying Gabor noise and regular Gabor noise. The hierarchical solution of spatially varying Gabor noise is only roughly a factor two slower than regular Gabor noise. We have also compared the performance of Perlin noise, regular Gabor noise and spatially varying Gabor noise in the specific case of figure 16. The performance of Perlin noise was 763 FPS, that of regular Gabor noise 22.8 FPS, and that of spatially varying Gabor noise 20.6 FPS (2 octaves, 1024×1024 , NVIDIA Quadro FX 3800 GPU). In this specific case, spatially varying Gabor noise is almost as fast as regular Gabor noise. Perlin noise is significantly faster than Gabor noise, but also lacks several attractive features of Gabor noise such as spatial variation and filtering.

A somewhat similar mechanism as the one used for spatially varying Gabor noise was recently used by Benard et al. [2010] to construct a dynamic noise primitive for coherent stylization in expressive rendering.

VI. CONCLUSION

In this paper, we have presented three significant improvements to Gabor noise: an isotropic kernel for Gabor noise, which speeds up isotropic Gabor noise with a factor of roughly two, an error analysis of Gabor noise, which relates the kernel truncation radius to the relative error of the noise, and spatially varying Gabor noise, which enables spatial variation of all parameters. These improvements build upon and augment the strong theoretical foundation of Gabor noise, and make Gabor noise an even more attractive alternative for existing noise functions.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers. We are grateful to the NVIDIA Professor Partnership Program for equipment donations. Ares Lagae is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO), and acknowledges K.U.Leuven CREA funding (CREA/08/017). Sylvain Lefebvre receives support from the Agence Nationale de la Recherche (SIMILAR-CITIES ANR-2008-COORD-021).

REFERENCES

- Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover, 1972. Ninth Dover printing, with corrections.
- Pierre Benard, Ares Lagae, Peter Vangorp, Sylvain Lefebvre, George Drettakis, and Joelle Thollot. A dynamic noise primitive for coherent stylization. *Computer Graphics Forum (Proceedings of the 20th Eurographics Symposium on Rendering)*, 29(4):1497–1506, 2010.
- H. R. Blackwell. Luminance difference thresholds. In D. Jameson and L. M. Hurvich, editors, *Handbook of Sensory Physiology*, volume VII/4, pages 86–99. Springer-Verlag, Berlin, 1972.
- Ronald N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 3rd edition, 2000. International editions 2000.
- Ronald N. Bracewell. *Fourier Analysis and Imaging*. Springer, 2004.
- Robert L. Cook and Tony DeRose. Wavelet noise. *ACM Transactions on Graphics*, 24(3):803–811, 2005.
- David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers, Inc., 3rd edition, 2002.
- A. Goldberg, M. Zwicker, and F. Durand. Anisotropic noise. *ACM Transactions on Graphics*, 27(3):54:1–54:8, 2008.
- Danny Holten, Jarke J. Wijk, Van, and Jean-Bernard Martens. A perceptually based spectral model for isotropic textures. *ACM Transactions on Applied Perception*, 3(4):376–398, 2006.

- Briggs Keith. Lambert W function. <http://keithbriggs.info/software/LambertW.c>, 2009.
- A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker. A survey of procedural noise functions. *Computer Graphics Forum*. to appear.
- Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse Gabor convolution. *ACM Transactions on Graphics*, 28(3):54:1–54:10, 2009a.
- Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse Gabor convolution - auxiliary material. Report CW 545, Department of Computer Science, K.U.Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium, May 2009b.
- J. P. Lewis. Algorithms for solid noise synthesis. In *Computer Graphics (Proceedings of ACM SIGGRAPH 89)*, volume 23, pages 263–270, 1989.
- Athanasios Papoulis and Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 4rd edition, 2002.
- Ken Perlin. Improving noise. In *Proceedings of ACM SIGGRAPH 2002*, pages 681–682, 2002.
- William H. Press, William T. Vetterling, Saul A. Teukolsky, and Brian P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, 2nd edition, 2002.
- Wim C. van Etten. *Introduction to Random Signals and Noise*. Wiley, 2005.
- Jarke J. van Wijk. Spot noise texture synthesis for data visualization. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, volume 25, pages 309–318, 1991.
- Colin Ware and William Knight. Using visual texture for information display. *ACM Transactions on Graphics*, 14(1): 3–20, 1995.
- Steven Worley. A cellular texture basis function. In *Proceedings of ACM SIGGRAPH 1996*, pages 291–294, 1996.
- Jainguo Zhang, Tieniu Tan, and Li Ma. Invariant texture segmentation via circular Gabor filters. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 901–904 vol.2, 2002.

APPENDIX I

CIRCULARLY SYMMETRIC FUNCTIONS

In this section we cover circularly symmetric functions. We review hyperspherical coordinates (subsection I-A) and the integration of circularly symmetric functions (subsection I-B), we simplify the convolution of circularly symmetric functions (subsection I-C), and we review the Hankel transform (subsection I-D).

A. Hyperspherical Coordinates

The hyperspherical coordinate system, the generalization of two-dimensional polar coordinates and three-dimensional spherical coordinates, is a natural coordinate system for working with circularly symmetric functions. The hyperspherical coordinates of a point in n -dimensional space with Cartesian coordinates (x_1, \dots, x_n) consist of a radial coordinate, r , and $n - 1$ angular

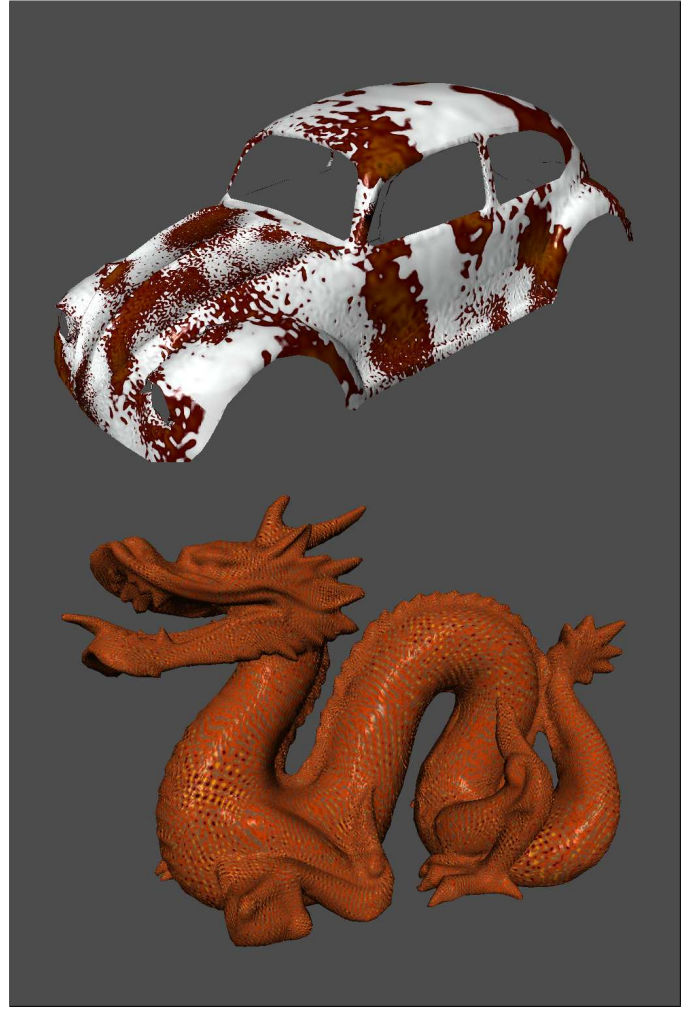


Fig. 16. Procedural textures generated with spatially varying surface Gabor noise. (top) A rusty car. (bottom) A dragon covered in scales. The spatial variation in the size of the rust patterns and the scales is steered by surface curvature.

coordinates, $\phi_1, \dots, \phi_{n-1}$, where

$$\begin{aligned} x_1 &= r \cos \phi_1 \\ x_2 &= r \sin \phi_1 \cos \phi_2 \\ &\vdots \\ x_{n-1} &= r \sin \phi_1 \dots \sin \phi_{n-2} \cos \phi_{n-1} \\ x_n &= r \sin \phi_1 \dots \sin \phi_{n-2} \sin \phi_{n-1} \end{aligned}, \quad (27)$$

where $r \in [0, \infty)$, $\phi_1 \dots \phi_{n-2} \in [0, \pi)$ and $\phi_{n-1} \in [0, 2\pi)$. The corresponding volume element is

$$r^{n-1} \sin^{n-2} \phi_1 \sin^{n-3} \phi_2 \dots \sin \phi_{n-2} dr d\phi_1 \dots d\phi_{n-1}. \quad (28)$$

B. Integration of Circularly Symmetric Functions

The integration of circularly symmetric functions can be simplified by exploiting their symmetry. When a function f is circularly symmetric in n dimensions, then the integral of f over \mathbb{R}^n reduces

to a one-dimensional integral,

$$\int_{x_1=-\infty}^{+\infty} \cdots \int_{x_n=-\infty}^{+\infty} f(x_1, \dots, x_n) dx_1 \cdots dx_n = S_n \int_{r=0}^{\infty} f(r) r^{n-1} dr, \quad (29)$$

where S_n is the hyper-surface area of an n -sphere of unit radius,

$$S_n = \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}, \quad (30)$$

where Γ is the Gamma function.

C. Convolution of Circularly Symmetric Functions

We simplify the convolution of circularly symmetric functions by exploiting their symmetry. First, we formulate the convolution of two n -dimensional circularly symmetric functions f and g using hyperspherical coordinates (see subsection I-A) as

$$[f * g](r) = \int_{r'=0}^{\infty} \int_{\phi'_1=0}^{\pi} \cdots \int_{\phi'_{n-2}=0}^{\pi} \int_{\phi'_{n-1}=0}^{2\pi} f(r') g(R) r'^{n-1} \sin^{n-2} \phi'_1 \sin^{n-3} \phi'_2 \cdots \sin \phi'_{n-2} dr' d\phi'_1 \cdots d\phi'_{n-1}, \quad (31)$$

where $R^2 = r^2 + r'^2 - 2rr' \cos \phi'_1$. We set the angular coordinates $\phi'_1 \cdots \phi'_{n-1}$ to zero in the expression for R^2 , since all functions are circularly symmetric, leaving only the angular coordinate ϕ'_1 . Next, we simplify the convolution using the integral

$$\int_0^{\pi} \sin^n \theta d\theta = \sqrt{\pi} \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n+2}{2})}, \quad (32)$$

where Γ is the Gamma function. Finally, we obtain

$$[f * g](r) = \frac{2\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n-1}{2})} \int_{r'=0}^{\infty} \int_{\phi=0}^{\pi} f(r') g(R) r'^{n-1} \sin^{n-2} \phi dr' d\phi, \quad (33)$$

with $R^2 = r^2 + r'^2 - 2rr' \cos \phi$. For $n = 2$, equation 33 corresponds to the equation given in Bracewell [2000, table 13.3].

D. The Hankel Transform

The Hankel transform [Bracewell, 2000, 13] [Bracewell, 2004, 9] is the method of choice for working with Fourier transforms of circularly symmetric functions.

When a function f is circularly symmetric in n dimensions, that is, when $f(x_1, \dots, x_n) = f(r)$, where $r^2 = x_1^2 + \dots + x_n^2$, then F , the n -dimensional Fourier transform of f , is also circularly symmetric, that is, $F(f_{x_1}, \dots, f_{x_n}) = F(f_r)$, where $f_r^2 = f_{x_1}^2 + \dots + f_{x_n}^2$. The relation between the one-dimensional functions $f(r)$ and $F(f_r)$ is given by the Hankel transform of order n . More specifically, the n -dimensional Fourier transform of $f(x_1, \dots, x_n)$ is given by the order- n Hankel transform of $f(r)$, that is, ${}^n\mathcal{F}[f(x_1, \dots, x_n)] = {}^n\mathcal{H}[f(r)]$. Note that, somewhat counterintuitively, the n -dimensional Fourier transform of $f(x_1, \dots, x_n)$ is not equal to the one-dimensional Fourier transform of $f(r)$, that is, ${}^n\mathcal{F}[f(x_1, \dots, x_n)] \neq {}^1\mathcal{F}[f(r)]$.

The Hankel transform, also called the Fourier-Bessel transform, is a one-dimensional integral transform with a Bessel function kernel. The Hankel transform of order n is

$${}^n\mathcal{H}[f(r)] = F(f_r) = \frac{2\pi}{f_r^{\frac{1}{2}n-1}} \int_0^{\infty} f(r) J_{\frac{1}{2}n-1}(2\pi f_r r) r^{\frac{1}{2}n} dr, \quad (34)$$

where J_n is the order- n Bessel function of the first kind. The Hankel transform is strictly reciprocal. An order- n Hankel transform pair is denoted as $f(r) \xleftrightarrow{{}^n\mathcal{H}} F(f_r)$. For $n = 1$, the Hankel transform corresponds to the Fourier transform, since circularly symmetric functions are real and even, and $J_{1/2}(x) = (2/\pi x)^{1/2} \sin x$ and $J_{-1/2}(x) = (2/\pi x)^{1/2} \cos x$.

APPENDIX II EQUATIONS FOR ISOTROPIC GABOR NOISE USING THE ISOTROPIC GABOR KERNEL

In this section, we provide equations for working with one-, two-, three- and four-dimensional isotropic Gabor noise using the isotropic Gabor kernel. We provide equations for the isotropic Gabor kernel in the spatial domain (subsection II-A) and in the frequency domain (subsection II-B), the integral of the isotropic Gabor kernel squared (subsection II-C), the envelope of the isotropic Gabor kernel (subsection II-D), and the radius of the truncated isotropic Gabor kernel (subsection II-E).

A. The Isotropic Gabor Kernel in the Spatial Domain

We obtain the one-, two-, three- and four-dimensional isotropic Gabor kernel in the spatial domain from equation 11, keeping into account that that $J_{1/2}(x) = (2/\pi x)^{1/2} \sin x$ and $J_{-1/2}(x) = (2/\pi x)^{1/2} \cos x$. The one-, two-, three- and four-dimensional isotropic Gabor kernel in the spatial domain is

$${}^1_I g(r) = K e^{-\pi a^2 r^2} 2 \cos(2\pi F_0 r), \quad (35)$$

$${}^2_I g(r) = K e^{-\pi a^2 r^2} 2\pi F_0 J_0(2\pi F_0 r), \quad (36)$$

$${}^3_I g(r) = K e^{-\pi a^2 r^2} \frac{2F_0}{r} \sin(2\pi F_0 r), \quad (37)$$

$${}^4_I g(r) = K e^{-\pi a^2 r^2} \frac{2\pi F_0^2}{r} J_1(2\pi F_0 r). \quad (38)$$

We illustrate the isotropic Gabor kernel in the spatial domain in figure 1(e) and in figure 2(e).

B. The Isotropic Gabor Kernel in the Frequency Domain

We obtain the one-, two-, three- and four-dimensional isotropic Gabor kernel in the frequency domain from equation 14, keeping into account that $I_{1/2}(x) = (2\pi x)^{-1/2} (e^x - e^{-x})$ and $I_{-1/2}(x) = (2\pi x)^{-1/2} (e^x + e^{-x})$. The one-, two-, three- and four-dimensional isotropic Gabor kernel in the frequency domain is

$${}^1_I G(f_r) = \frac{K}{a} \left(e^{\frac{-\pi}{a^2}(f_r - F_0)^2} + e^{\frac{-\pi}{a^2}(f_r + F_0)^2} \right), \quad (39)$$

$${}^2_I G(f_r) = \frac{2\pi K F_0}{a^2} e^{\frac{-\pi}{a^2}(f_r^2 + F_0^2)} I_0\left(\frac{2\pi F_0}{a^2} f_r\right), \quad (40)$$

$${}^3_I G(f_r) = \frac{K F_0}{a f_r} \left(e^{\frac{-\pi}{a^2}(f_r - F_0)^2} - e^{\frac{-\pi}{a^2}(f_r + F_0)^2} \right), \quad (41)$$

$${}^4_I G(f_r) = \frac{2\pi K F_0^2}{a^2 f_r} e^{\frac{-\pi}{a^2}(f_r^2 + F_0^2)} I_1\left(\frac{2\pi F_0}{a^2} f_r\right). \quad (42)$$

We illustrate the isotropic Gabor kernel in the frequency domain in figure 1(f) and in figure 2(f).

C. The Integral of the Isotropic Gabor Kernel Squared

We obtain the integral of the isotropic Gabor kernel squared by integrating the Gabor kernel squared in the spatial domain or

in the frequency domain. The integral of the one-, two-, three- and four-dimensional isotropic Gabor kernel squared is

$$2 \int_{r=0}^{\infty} {}_1g^2(r) dr = \frac{\sqrt{2}K^2}{a} \left(1 + e^{-\frac{2\pi F_0}{a^2}}\right) \quad (43)$$

$$2\pi \int_{r=0}^{\infty} {}_2g^2(r) r dr = \frac{2\pi^2 K^2 F_0^2}{a^2} e^{-\frac{\pi F_0^2}{a^2}} I_0\left(\frac{\pi F_0^2}{a^2}\right) \quad (44)$$

$$4\pi \int_{r=0}^{\infty} {}_3g^2(r) r^2 dr = \frac{2\sqrt{2}\pi K^2 F_0^2}{a} \left(1 - e^{-\frac{2\pi F_0}{a^2}}\right) \quad (45)$$

$$2\pi^2 \int_{r=0}^{\infty} {}_4g^2(r) r^3 dr = \frac{2\pi^3 K^2 F_0^4}{a^2} e^{-\frac{\pi F_0^2}{a^2}} I_1\left(\frac{\pi F_0^2}{a^2}\right). \quad (46)$$

D. The Envelope of the Isotropic Gabor Kernel

We define the envelope of the isotropic Gabor kernel as the product of the Gaussian with the envelope of the harmonic. The envelope of $\cos(x)$ and $\sin(x)$ is 1, and the envelope of $J_n(x)$ is $M_n(x) = (J_n^2(x) + Y_n^2(x))^{1/2}$, where Y_n is the order- n Bessel function of the second kind. The envelope of the one-, two-, three- and four-dimensional isotropic Gabor kernel is

$${}_1g_E(r) = K e^{-\pi a^2 r^2} \quad (47)$$

$${}_2g_E(r) = K e^{-\pi a^2 r^2} 2\pi F_0 M_0(2\pi F_0 r) \quad (48)$$

$${}_3g_E(r) = K e^{-\pi a^2 r^2} \frac{2F_0}{r}, \quad (49)$$

$${}_4g_E(r) = K e^{-\pi a^2 r^2} \frac{2\pi F_0^2}{r} M_1(2\pi F_0 r). \quad (50)$$

The envelope of the isotropic Gabor kernel is illustrated in figure 6.

E. The Radius of the Truncated Isotropic Gabor Kernel

We obtain the radius of the truncated isotropic Gabor kernel by solving the envelope of the isotropic Gabor kernel for the radius, and evaluating the result for a specific value of the envelope. The envelope of the one-dimensional isotropic Gabor kernel solved for the radius is

$$r = \sqrt{\frac{-\log\left(\frac{{}_1g_E}{2K}\right)}{\pi a^2}}. \quad (51)$$

We solve the envelope of the two-dimensional isotropic Gabor kernel for the radius by approximating $M_n(x)$ by $M_n(x) \approx (2/\pi x)^{1/2}$, using the asymptotic expansions for large arguments [Abramowitz and Stegun, 1972, 9.2.1, 9.2.2]. The envelope of the two-dimensional isotropic Gabor kernel solved for the radius is

$$r \approx \frac{1}{2\sqrt{\pi}a} \sqrt{W\left(\frac{64\pi K^4 F_0^2 a^2}{x^4}\right)}, \quad (52)$$

where W is the Lambert W-function, that is, the inverse of $f(W) = W e^W$. The envelope of the three-dimensional isotropic Gabor kernel solved for the radius is

$$r \approx \frac{\sqrt{2}}{2\sqrt{\pi}a} \sqrt{W\left(\frac{8\pi K^2 F_0^2 a^2}{{}_3g_E^2}\right)}. \quad (53)$$

The envelope of the four-dimensional isotropic Gabor kernel cannot be solved for the radius.



Ares Lagae is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO), working at the Computer Graphics Research Group of the K.U.Leuven. He received the MS and BS degrees in computer science from the K.U.Leuven in 2000 and 2002. He received the PhD degree in computer science from the K.U.Leuven in 2007, working with Philip Dutré. He spent the academic year 2009 - 2010 at REVES, INRIA Sophia Antipolis - Méditerranée, working with George Drettakis. Since 2007, he is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO). More details about his research can be found at <http://people.cs.kuleuven.be/~ares.lagae/>.



Sylvain Lefebvre is a researcher at INRIA, France. He completed his PhD in 2004, on the topic of texturing and procedural texture generation using GPUs. After graduating he joined Microsoft Research (Redmond, USA) as a postdoctoral researcher during the year 2005. His current research focuses are in procedural content generation, end-user content manipulation and compact data structures for interactive applications and games. Sylvain received the EUROGRAPHICS Young Researcher Award in 2010 for his work on texture synthesis and GPU data-structures. More details about his research can be found at <http://www.loria.fr/~slefebvr>



Philip Dutré (1966) is full professor at the Dept. of Computer Science at the Katholieke Universiteit Leuven. His research interests include real-time computer graphics, photo-realistic image synthesis, procedural modeling primitives and perceptual heuristics for image generation. He has served on various program committees such as SIGGRAPH and EUROGRAPHICS, and served as Papers co-chair for EUROGRAPHICS 2009. Currently, he also is a member of the Executive Committee of Eurographics. He is the author of the book 'Advanced Global Illumination', with Kavita Bala and Philippe Beqaert.