Procedural Texture Synthesis by Locally Controlled Spot Noise

Nicolas Pavie[1], Guillaume Gilet[1], Jean-Michel Dischler[2], Djamchid Ghazanfarpour[1]

[1] XLIM - UMR CNRS 7252, University of Limoges, France

[2] ICUBE - UMR CNRS-UDS 7357, University of Strasbourg, France

ABSTRACT

Procedural noises based on power spectrum definition and random phases have been widely used for procedural texturing, but using a noise process with random phases limits the types of possible patterns to Gaussian patterns (i.e. irregular textures with no structural features). Local Random Phase (LRP) Noise has introduced control over structural features in a noise model by fixing the frequencies and phase information of desired features, but this approach requires storing these frequencies. Space distortion and randomization must also be used to avoid repetitions and periodicity. In this paper, we present a noise model based on non-uniform random distributions of multiple Gaussian functions for synthesizing semi-structured textures. We extend the LRP noise model by using a spot noise based on a controlled distribution of kernels (spots), as an alternative formulation to local noises aligned on a regular grid. Spots are created as a combination of Gaussian functions to match either a specific power spectrum or a user-defined texture element. Our noise model improves the control over local structural features while keeping the benefits of LRP noise.

Keywords

Procedural texturing, Image synthesis, procedural noise

1 INTRODUCTION

Random signals have been widely used for procedural texturing since the marble pattern of Perlin [Per85]. Noise-based procedural textures inherit many properties of procedural noises, the most compelling ones being :

- No repetition is visible;
- The pattern produced is continuous over its evaluation space;
- It can be computed during rendering on a per-pixel basis;
- One texture model can produce various patterns by tuning parameters

These advantages have led to a growing study of noise applications in procedural texturing. A large variety of patterns can be produced by a noise-based process by defining a given power spectrum (Gabor noise [LLDD09], Multiple Kernel noise [GDG12b]) but shaping a pattern by directly tuning the spectrum of a noise remains a difficult task, because the correlation between a target pattern and the corresponding power spectrum cannot be straightforwardly deduced. A more artist-friendly approach consists in computing the noise parameters from a pattern sample [GLLD12].

Most of the recent "noise by example" methods consider a given image as an input and aim at generating a noise reproducing its power spectral density (PSD), computed by spectral analysis. Visual variety in the results is introduced by keeping the phase information random. However, it is well known that structure can be found in the phase information of the spectral analysis [OL81].

The recently introduced LRP Noise [GSV*14] tackles the problem of structural features preservation by fixing both phases and magnitudes in some areas of the frequency spectrum approximating the structural component of the input example. This approach has two drawbacks. Firstly, it requires to store the phase information of the relevant parts of the spectrum. Secondly, as only a limited number of fixed frequencies are used, periodicity of the structural components must be broken by using turbulence [Per85] and random shifts.

In this paper, we present an alternative formulation of the LRP noise based on a locally defined and controllable spot noise representation. We focus on structural features that can be defined by a repetitive structured kernel function (i.e. fabric textures with specific stitches aspects and random small variations). Such kernels can be created by an arrangement of the base components of the features (i.e. the threads within a stitch). This formulation retains the advantages of the

LRP noise : Both stochastic details and structural features can be generated in real-time on a per pixel basis. Simple structures can be reproduced through an automatic process. The benefit provided by our representation is two-fold :

- The sum of cosines modeling the structural component of the LRP Noise formulation is replaced by a sum of few Gaussian kernels. This compact representation reduces computation cost for local structures and yields a noise process with similar performance but with enhanced control over the final visual appearance.

- The distribution of the structural features of the input can be edited and is part of the definition of the model. This extends the range of possible patterns that can be produced, from very regular to completely stochastic ones, but still featuring structural components. Repetitions and periodicity are avoided for semi-regular patterns since the distribution of local noises is still based on a random process.

The control over both distribution and kernel aspect in the noise model allows for interactive edition of patterns.

A comparison of local noise formulations is presented in fig.1.We present the possibilities offered by our new formulation through several examples of patterns as shown in fig.3.

2 RELATED WORKS

2.1 Procedural patterns synthesis

To create procedural patterns, several approaches can be used depending on the desired degree of "randomness". For structured and semi-structured procedural patterns, patch-based approach can help artists expanding a pattern sample with characteristic structural features. In such approach, a procedural pattern is evaluated by tiling the surface with patches (small textures) [CSHD03, EF01, VSLD13]. Patches are randomly arranged to break repetitions, but results may lack of details variety : the same tiles / patches (i.e. rigorously identical contents) are repeated over and over again even for irregular textures .

Semi-structured pattern can also be synthesized as a distribution of objects in texture space [GD10]. To create a procedural pattern with this approach, a procedural distribution function is required to create an infinite set of random position. Point jittering is often used as distribution function for its simplicity and evaluation speed [Gla04]. But it does not take in account spatial dependencies (distance threshold between objects) so distributed objects may overlap. *Direct Stochastic tiling*

[LD05] can produce some distance dependencies, to create for example an infinite set of Poisson-disks. But it still requires some tiles to be precomputed and stored. For their *assemblage* creation, [GDG12a] proposed an improvement of point jittering to take in account some spatial dependencies : the squared lattice is replaced by polygonal cells that forms a rectilinear tesselation of the plane. Similarly to jittering, each cell contains a different instance of an object with a random position computed on-the-fly. Fully procedural semi-structured pattern can be produced using both procedural objects definition and procedural distribution function, but very few techniques propose to extract such objects directly from an input sample. Irregular and near-regular patterns can also be generated with Markov Random Fields [CJ83]. [VGR16] specifically consider Markov-Gibbs Random Fields to create stochastic, irregular and near-regular textures. This approach can reproduce patterns with complex structural details from an example with great accuracy. But the texture generation processes associated with such models are highly iterative and focus on statistical reproduction over generation speed. It makes them unpractical to use in a rendering pipeline for high resolution textures generation on-the-fly.

To create procedural pattern with greater randomness, procedural noise functions are often preferred over spatial description methods (more details in section 2.2). But as modeling a power spectrum is no trivial exercise, several noises "by example" use a self-configuration process to approximate a specified power spectrum, within the noise spectral capabilities. [LVLD10] describe a process to reproduce isotropic patterns by decomposing a Power Spectral Density (PSD) into several frequency bands to compute the weights of a multi-resolution wavelet noise. [GDG12b] extract several kernel configurations from an arbitrary PSD by decomposing a spectral domain into sub-regions of specific magnitude range. As an extention of the Gabor noise, [GLLD12] also describe a method to reproduce an arbitrary PSD in several band-limited Gaussian spectrums. Each spectrum corresponds to a band limited Gabor noise. These noises are nonetheless limited to Gaussian patterns : as they are completely characterized by their power spectrum, only micro-structural features are produced. Local Random Phase noise [GSV*14] is of particular interest as it introduces structure preservation in its noise formulation while allowing the "by example" approach.

2.2 Procedural noises

Procedural noises have been widely used as a modeling tool for texture synthesis after the Perlin noise first appeared in [Per85]. A procedural noise implies no discrete data samples, a very low storage requirement (i.e. a simple evaluation function), no periodicity nor repetitions. Two families of procedural noises are gener-

ally considered (see survey [LLC*10]) : lattice gradient noises and sparse convolution noises. Lattice gradient noises are based on the interpolation of randomly oriented gradient ([EMP*02]) dispatched on a regular grid. Sparse convolution noises are based on the convolution of a spatial filter function (kernels) with a random distribution of impulses (points).

Random distribution processes result in a white noise in the frequency domain, so the control of sparse convolution noises can be achieved by spectral definition of the kernel function. A sparse convolution noise can be constructed around a specific evaluation functions such as Gabor-[LLDD09], Gaussian-[Lew89], or Sync-[GDG12b] kernels. The latter use multiple configurations of the kernel to optimize spectral coverage. A more spatial-oriented formulation of a sparse convolution noise was proposed by [vW91, dLvL97] with the Spot Noise. It is based on an arbitrary spatial kernels. Some micro-structural features can be produced by using structured kernel. But the quantity of the structural features produced remains limited by the random distribution process.

Local Random Phase noise [GSV*14] states that structural features are contained in both the magnitude and the phase spectrum of specific frequencies. To produce structures within a noisy pattern, LRP noise model propose to fix their corresponding frequencies. While it achieves to produce structures accurately, this noise model suffer from several drawbacks : 1) frequencies of structures selected for reproduction need to be stored; 2) Local cosine-based noises need a great number of cosines to cover the spectrum.

We extend the LRP noise formulation to produce a more compact representation of local structures by relying on a spot noise formulation. Our local spot noises use a sum of quadratic Gaussian functions to create structured or unstructured kernels, so a wide range of possible spot aspects can be produced. Locally defined structural features are further enhanced by the introduction of a constrained random distribution.

3 NOISE MODEL

We now present our alternative formulation of the LRP noise model based on spot noise. As a reminder, the original formulation of the LRP noise is the following

$$n(x) = \sum_{i=1}^{I} w\left(\frac{||x - x_i||}{\Delta}\right) \sum_{j=1}^{J} A_{i,j} \cos(2\pi f_{i,j} \cdot x + \rho_{i,j})$$

(1)

It is a mix of $J \times I$ local cosine-based noises with random phases and windowed over a regular lattice (x_i is the position over the spatial lattice corresponding to a spectral stratum i) : randomness is obtained by the random phases while the spectrum is controlled by the frequencies sampling of each noise. We now propose an

alternative formulation based on a fully procedural spot noise to limit the number of cosines. We also present a new procedural distribution function to control the locality of the noise produced. The extended range of patterns that can be produced using this distribution is presented in fig. 3.

3.1 Procedural Multiple-Gaussian spot noise

Sparse convolution noise [Lew89] is originally based on the random distribution of impulses convolved with an isotropic Gaussian kernel. Such kernels, created as the multiplication of a sample texture by a Gaussian envelope, only produce isotropic Gaussian patterns due to the fixed Gaussian envelope of the kernel used. To improve spectral control, Gabor kernel [LLDD09] can be used as it unifies spectral and spatial characteristics. But spatial control is reduced at the same time. It can produce only Gaussian textures, which is an excessively narrow subset of procedural patterns. Spot noise [dLvL97] can produce a wider range of patterns, including non-gaussian patterns containing structural features, by using an arbitrary spatial kernel instead.

We aim at spatial characteristics that cannot be produced by the sole power spectrum definition. [vW91] noted that structural characteristics present in the kernel itself, such as (an)isotropy or a structural feature, result in similar characteristics within the texture produced by the spot noise. In other words, when the kernel contains some structure, this structure is transferred to the texture. A formulation of spot noise is given by [dLvL97] as :

$$n_s(\mathbf{p}) = \sum_{j}^{J} w_j(\mathbf{p}_j) k_s\left((\mathbf{p} - \mathbf{p}_j)\mathbf{R}_s(\mathbf{p}_j)\mathbf{S}_s(\mathbf{p}_j)\right)$$

(2)

Where $\mathbf{p}_j$ is an impulse position, and k_s is the kernel function of the spot s. Orientation $\mathbf{R}_s$ and scale $\mathbf{S}_s$ are related to underlying data fields. Impulse positions $\mathbf{p}_j$ are uniformly distributed using a Poisson process and the weights w_j are equiprobably drawn in $[-1, 1]$.

For our kernel formulation, we use a sum of ellipsoidal N-dimensional Gaussian functions with arbitrary scale and orientation. Gaussian functions (and kernels by extension) are commonly used in noise literature to create Gaussian noise patterns. It has also been used for modeling surfaces and volumes [JBL*06]. In computer vision, Gaussian kernels can be used as a reconstruction primitive after a Gabor-wavelet decomposition of an image [WM03]. For spot noise modeling, a wide range of kernels can be produced by combining a few simple Gaussian functions. Some examples of kernels are presented in figure 2. Our kernel k is defined for a dimension N as:

$$k(\mathbf{p}) = \sum_{V_i} g_{V_i} \qquad g_V(\mathbf{p}) = A e^{-\frac{1}{2}\mathbf{p}^T \mathbf{V}^{-1} \mathbf{p}}$$

(3)

A) Gaussian noise
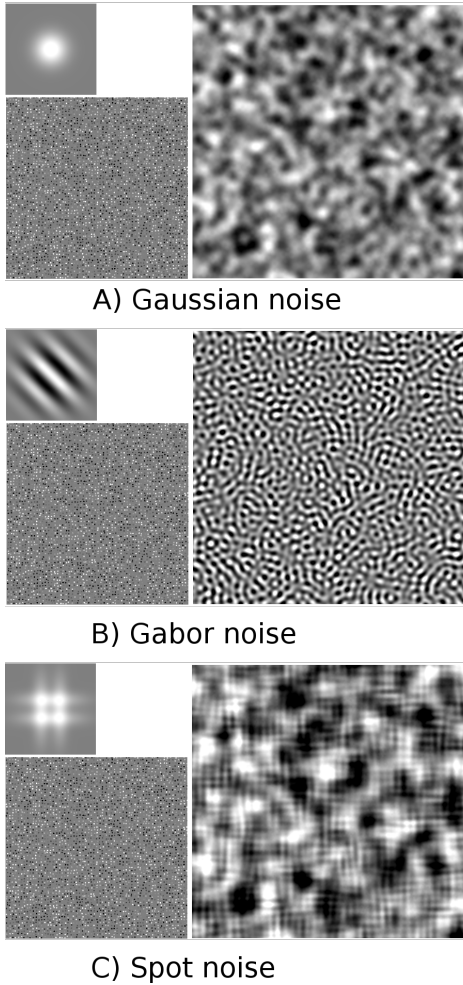

B) Gabor noise

C) Spot noise

Figure 1: Examples of noises generated by sparse convolution, with their respective kernel and impulse distribution. Unlike Gaussian (a) and Gabor (b) noise, spot noise (c) can also produced semi-regular structural features using a single arbitrary spatial kernels.
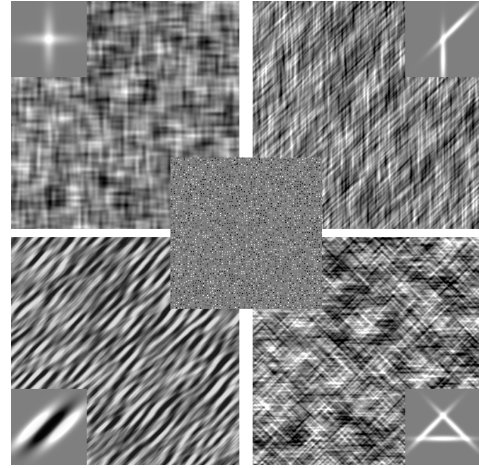

Figure 2: Spot noise with various kernel aspects, composed of several elliptic Gaussian function. Each oriented Gaussian within a kernel generates an oriented component within the results. The center tile shows their random distribution of impulses. Performances are around 85 fps for the 2-Gaussians spots (top row) and 65 fps for the 3-Gaussians spots (bottom row).

Here A is the Gaussian magnitude, and $\mathbf{V}$ is a $(N+1) \times (N+1)$ matrix, such that $\mathbf{V}^{-1} = (\mathbf{M\,R\,S})^{-T}(\mathbf{M\,R\,S})^{-1}$ and $|\mathbf{V}|$ is the matrix determinant. $\mathbf{M}, \mathbf{R}$ and $\mathbf{S}$ respectively correspond to shift, rotation and scaling matrices. The isocountour of the kernel is given by $\mathbf{p}^T\mathbf{V}^{-1}\mathbf{p}$, which describes an implicit surface given that $\mathbf{V}$ is a semi-positive matrix ($\mathbf{p}$ is a point in dimension $N+1$ with last component set to 1). We show in figure 1.c an example where using a simple grid-kernel composed of four ellipsoidal Gaussian can effectively produce a texture with semi-regular structural features. With the sparse convolution process, the evaluation window is considered to be induced by the kernel formulation (i.e. each Gaussian function falls below a threshold before the maximum evaluation distance of a kernel). Consid-

ering this and equation 2, the new formulation of our complete noise model becomes :

$$n(\mathbf{p}) = \sum_{i=0}^{I} w_i n_i(\mathbf{p}) \qquad (4)$$

Where n_i is a local noise composed of random impulses of the kernel i. Here w_i is an energy normalization factor computed for each noise. Our noise model is a composition of several spot noises. Each spot noise can be used to model a specific set of features of a pattern.

Texture generated by this formulation produces local structural features (see the "grid-shaped" micro-pattern in figure 1(c)) while keeping the randomness introduced by the Poisson distribution of impulses. To further widen the range of possible patterns generated by this formulation, we propose to extend the previous spot noise by introducing a non-uniform random distribution of impulses.

Let $\mathbf{p} = (X,1)$, a point of coordinates X in a specified dimension, we propose a new formulation :

$$n_s(\mathbf{p}) = \sum_j \delta\left(\xi(\mathbf{p}_j) < d(\mathbf{p}_j)\right) |w_j(\mathbf{p}_j)| K_j(\mathbf{p}) \qquad (5)$$

with $K_j(\mathbf{p}) = k_s\left((\mathbf{p}-\mathbf{p}_j)\,\mathbf{R}_s(\mathbf{p}_j)\,\mathbf{S}_s(\mathbf{p}_j)\right)$. d is a scalar field and represents a probability. δ denotes the Kronecker delta and $||$ the absolute value. ξ is a random variable selected independently of w_j. d allows us to control the density of impulses in given regions. Because we use an absolute value, high density regions imply noise values close to 1 whereas low density values result in values close to 0.
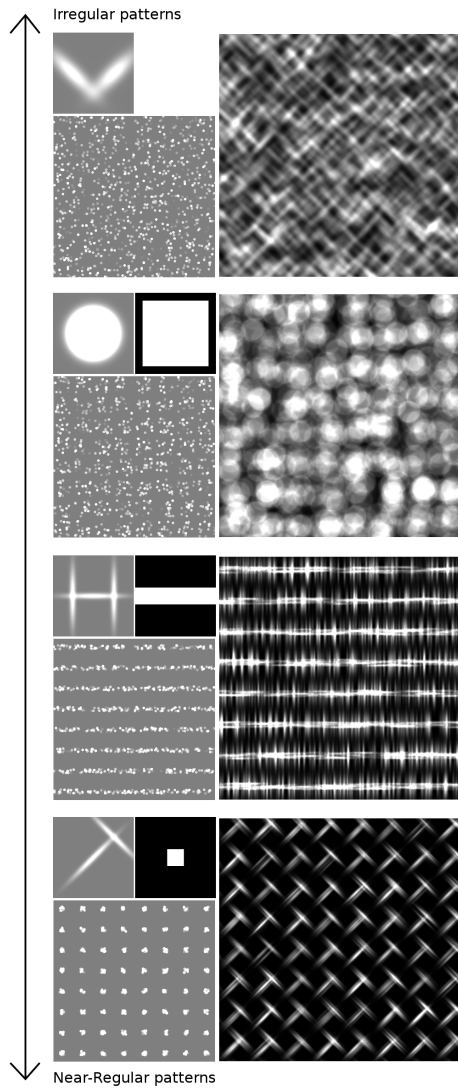
Irregular patterns

Near-Regular patterns

Figure 3: Examples of pattern produced with our distribution, from irregular (top row) to regular patterns. For each example, the top images of the the left column are the kernel (left) and the periodic density (right) profiles used, the bottom left image shows the resulting impulse distribution. The image of the right column shows the noise result.

The global appearance of the produced texture is directly correlated to the shape of d : the energy of the pattern is concentrated around higher density areas within the density field. It introduces a new level of control over the various appearance of the generated texture and can be used to introduce global structure at a large scale. To model structural regularity, we define d as a periodic density field tiling the evaluation space (i.e. fig. 3,6,7), or as a global density field (i.e. fig. 8). For convenience, periodic density fields used in this paper are represented by the density for a single period and referred as density or distribution profiles. The profiles are created using simple shapes functions

to allow interactive authoring and fast evaluation. Figure 3 demonstrates the range of appearance produced by this control over the distribution of impulses using different periodic density profiles. Our formulation encompasses previous noise formulations and is thus able to produce various patterns, from irregular patterns (by using a random distribution, cf. figure 3 top) , to near-regular patterns(cf. figure 3, bottom).

Note that in spite of regularity in global appearance, this texture preserves randomness : impulses are still generated using a random distribution process: only the density of impulses varies spatially. We experienced that the shape of density profiles allows an easy and intuitive control of texture structure : structural alignments result in small irregular gaps or aligned variations in the pattern. Such insights can be used to visually estimate the elements positioning margin within an area and to recreate the density profile accordingly. The user can also edit kernels in real-time, getting an instant texture feedback while creating or modifying the pattern.

4 BY EXAMPLE PROCEDURAL TEXTURING

As shown in section 3, our spot noise model is well suited to manage some types of structural features. We have further shown that the structure can be represented either by the kernel itself or by the distribution of impulses (using non-uniform distributions). However analyzing the input texture to obtain both the distribution and the shape of the kernel is a difficult and challenging task: both are strongly linked and they can only hardly be decoupled. A solution consists in fixing one of the two. In this paper we propose to use a similar approach as for LRP noise: we extract complex kernels from examples that are then distributed over a regular grid.

First, we briefly present a summary of the LRP Noise by example approach introduced to process some types of structured textures.

Summary of by-example LRP Noise

By-example LRP Noise is based on a spectrum segmentation to extract the magnitudes and phases of structural features : the input spectrum is stratified according to energy levels, and then subdivided in sub-strata to compute local noises.

A stratum R corresponding to the highest energy area in the power spectrum is considered as defining the structure of the pattern. This region is seen as the frequencies "containing the structure" and is chosen by a tunable parameter $r \in [0;1]$ such that the proportion of total energy contained within R is r. The most important

structural features are preserved by fixing the phases and amplitudes over the corresponding frequencies. At the extremities, the resulting texture varies from a fully procedural ($r = 0$) to a copy of the original sample ($r = 1$). In practice and for weakly structured random textures, authors report a value of $r \approx 20\%$ as an efficient value for preserving both the structural features and the randomness of the pattern. Final noise (texture) computation is done by summing the noises approximating all energy strata :

$$n = n_R + \sum_S n_S \qquad (6)$$

Two types of noises are thus considered : noises n_S relying completely on power spectrum and purely random phases (they keep storage requirements minimal), while the noise n_R has fixed phases and amplitudes.

One drawback is that a high amount of cosine waves are needed to accurately represent n_R, thus generating an important computational overhead. Gilet et al. [GSV*14] deal with this issue by trading continuity for computational efficiency. Basically, the structure image (the inverse Fourier transform of n_R) is iteratively decomposed into a regular grid of blocks. A block-wise FFT of this structure image is computed and a fixed amount of highest-amplitude frequencies are selected and stored for each block. n_R is finally evaluated by block in the spatial domain and re-assembled during rendering using the windowing function. We refer the reader to [GSV*14] for more details about LRP Noise by example.

4.1 Reproducing structure with Locally Controlled Spot Noise

First we have to separate the input texture into the structural part and the Gaussian random part. To this end, we propose to use the same technique as for the LRP Noise method, i.e. to consider a *structure* image constructed as the inverse Fourier transform of the highest energy region of the spectrum. The goal of our method is to compute a collection of I local kernels, each encoding a part of the structure. We achieve this by subdividing the structure image following a regular grid of arbitrary resolution and computing a Gaussian-based representation for each of the resulting blocks.

This ends up in computing a compact elliptical Gaussian-based representation of a given image, which is a difficult process when images are complex. By using a standard ellipse fitting algorithm, such as the method proposed in [AWF95], the pixels of the input image are approximated by J ellipses, which are then expressed as elliptical Gaussian functions of
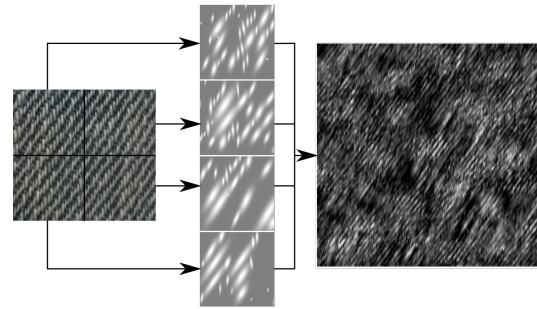

Figure 4: Noise by example : an input sample (left image) is subdivided into smaller samples. Each sub samples is decomposed in Gaussian functions (around 20 for this example), giving for each sub-samble a specific spot (center). Resulting pattern (right) is obtained by random distribution of the spots. Performances are about 10 fps.

corresponding radius. As illustrated in figure 4, these functions are the basis of our local spot noise kernel. The efficiency of ellipse fitting is strongly depending on the complexity of the image and is able to work only on simple features such as presented in figures 4 and 5. A deeper analysis and segmentation of the image could further lead to an increase of the quality of the approximation and could in future work allow the reproduction of more complex structural features.

The number J of Gaussian functions is constant for all blocks of the image and impact the accuracy of the approximation of each structural feature and the performance of the spot-noise during rendering. The rendering speed is linearly dependant on the number of Gaussian functions composing each kernel. This trade-off between accuracy and performance is a parameter of our model and chosen by the user. In practice, all results in this paper are computed with J between 4 and 8, and up to 20 for very complex spots.

4.2 Combining structure and noise

During rendering, the impulses are distributed using jittering (random displacement of points defined on the integer lattice). The resolution of the lattice corresponds to the resolution used during the subdivision process of the structure image. Each impulse is associated with a kernel, that can be chosen as the kernel approximating the block (in the structure image) corresponding to the integer lattice of the impulse or randomly chosen to increase randomness. By using the kernel approximating the corresponding structure block, low frequency structural features can be represented by the combination of kernel across the output, at the cost of the randomness.

The Gaussian random part of the input texture is then added by a cosine-based kernel noise as in the standard

LRP Noise method or as a random distribution of simple kernels defined as in figure 1.

5 RESULTS

We implemented our noise as a GPU fragment shader using OpenGL. Random numbers were generated by a linear congruential PRNG initialised by a Morton coded seed similarly to [LLDD09]. Performances are strongly dependant on the impulses density and kernels complexity. All results in this paper are rendered between 10 and 165 fps in a 1200×1200 window on a GeForce 980.

Figures 4 and 5 shows examples of structure reproduction obtained from an input example. As can be seen, the simple shape of the input structural feature is accurately reproduced by our automatic process. As stated earlier, our method relies on automatic segmentation and computation of Gaussian representation of an input pattern. Using a straightforward ellipse fitting technique provides results for simple patterns but automatic analysis of a complex pattern remains a difficult challenge. We however believe that this is a first step toward fully automatic by-example procedural texturing of complex patterns using locally controlled spot noises.
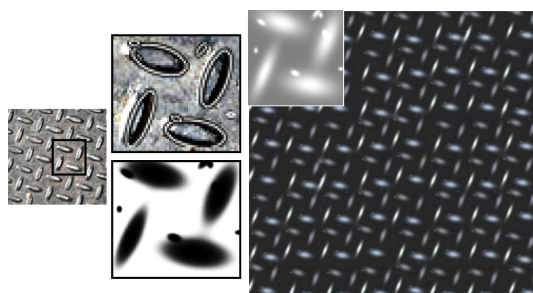

Figure 5: A simple example of pattern modeling obtained from the input example (left). The shape of the structural pattern is computed by ellipse fitting and expressed as a sum of Gaussian functions (middle) to produce the final structure (right). Performances are around 165 fps.

Several patterns can still be represented using an user provided kernel. Figure 6 shows several examples of pattern reproduction through a given periodic distribution and an adequate user-defined kernel.

Unlike recently introduced noise methods, our technique focus on the edition of pattern in the spatial domain. Indeed, interactive modeling in the spatial domain is easier than the direct edition of a power spectrum. Figure 7 illustrates our edition pipeline and shows how the kernel and distribution can be edited to impact the global structure of the target pattern with instant feedback for the user. Figure 8 illustrates the control capabilities of our noise model over multiple structures distribution within a single pattern. Figure 9 illustrates
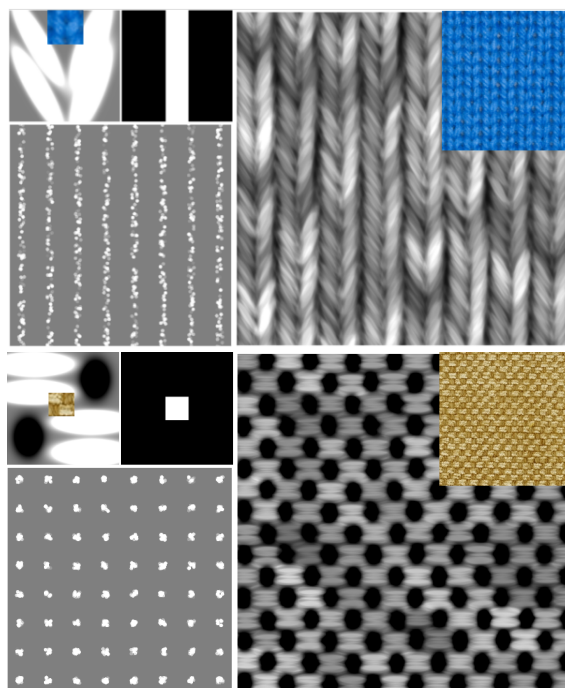

Figure 6: Examples of a near-regular features reproduction by a single spot noise. The kernel profile (left column, top left profile) is provided as a user-guided Gaussian decomposition of a texture element (i.e. a mesh of fabric). The distribution profile (left column, top right profile) is either directly authored or provided as a small texture. The top right tiles show the original pattern from which the sample to reproduce as a kernel were extracted. Performances are around 60 fps for the top pattern and 40 fps for the bottom pattern.

an application of a pattern on a 3D object with a simple bump mapping. This figure uses the top row configuration of the figure 6 to compute a noise used as a height field. Normals used for the bump mapping were computed by finite differences over 3 evaluations of the noise in a fragment shader.

6 CONCLUSION

We have introduced a new noise model based on locally controlled spot noises to reproduce from near-regular to irregular pattern features. Near-regular features are produced by combination of structured kernel and a controlled random distribution process. As it extends the Local Phase Noise model, it can still reproduce irregular patterns with structural features.

Our noise function contrasts with most recent research papers concerning noise models because our focus is not to match a given power spectrum, but rather to focus on spatial structure control : sculpting interactively a pattern shape in spatial domain is an easier creation process than editing a given power spectrum. Noise is hard to control, and generally ill suited for the modeling of structured procedural textures. We believe that
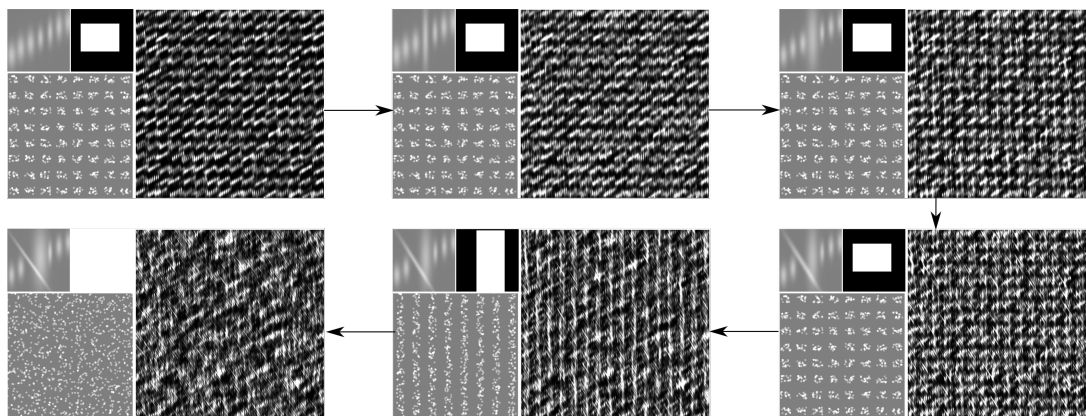
Figure 7: An example of edition process. Giving an input spot and distribution (top left), the user can interactively modify each Gaussian function of the spot and the distribution profile to change the appearance of the result. Performances are around 45 fps.
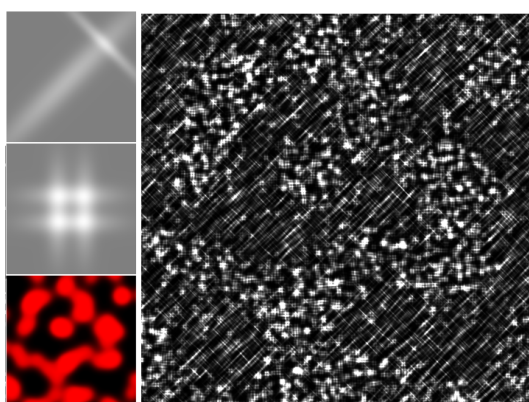

Figure 8: An example of mixed kernels for structures repartition within a single noise pattern. A global density field (bottom left), generated by a secondary spot noise, is used to control the distribution of spots : for an impulse distributed, the corresponding spot (top left profile or middle left profile) is selected according to a random density test. Performances are around 55 fps.

our locally controlled spot noise could provide a first hint to address the difficult problem of modeling structured patterns. In particular, one important extension for future work would be the exploration of an automatic example-based method. Such a method would use as input a photograph of surface details and then attempt to derive a corresponding set of kernels and impulse density distributions.

7 REFERENCES

[AWF95] ANDREW W. FITZGIBBON R.: A buyer's guide to conic fitting. In *Proc. 5th British Machine Vision Conference, Birmingham* (1995), pp. 513–522.

[CJ83] CROSS G. R., JAIN A. K.: Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Intell. 5*, 1 (Jan. 1983), 25–39.
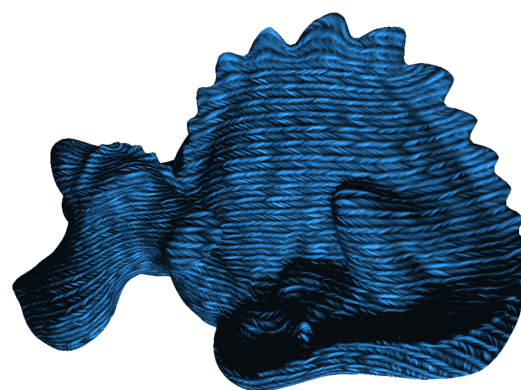

Figure 9: An example of the blue fabric pattern (fig. 6, top row) applied on a 3D model. The noise function was used to compute heights. Normals of the resulting pattern were computed using finite differences and used in a simple bump mapping process. Performances are around 20 fps.

[CSHD03] COHEN M. F., SHADE J., HILLER S., DEUSSEN O.: Wang tiles for image and texture generation. *ACM Trans. Graph. 22*, 3 (July 2003), 287–294.

[dLvL97] DE LEEUW W., VAN LIERE R.: Divide and conquer spot noise. In *Proceedings of the 1997 ACM/IEEE Conference on Supercomputing* (New York, NY, USA, 1997), SC '97, ACM, pp. 1–13.

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 341–346.

[EMP*02] EBERT D. S., MUSGRAVE F. K., PEACHEY D., PERLIN K., WORLEY S.: *Texturing and Modeling: A Procedural Approach*, 3rd ed. Morgan Kaufmann Publishers Inc., San

Francisco, CA, USA, 2002.

[GD10] GILET G., DISCHLER J. M.: Procedural texture particles. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, ACM, pp. 6:1–6:1.

[GDG12a] GILET G., DISCHLER J.-M., GHAZANFARPOUR D.: Multi-scale assemblage for procedural texturing. *Computer Graphics Forum 31*, 7 (2012), 2117–2126.

[GDG12b] GILET G., DISCHLER J.-M., GHAZANFARPOUR D.: Multiple kernels noise for improved procedural texturing. *The Visual Computer 28*, 6 (2012), 679–689.

[Gla04] GLANVILLE S.: Texture bombing. In *GPU Gems*. 2004, ch. 20.

[GLLD12] GALERNE B., LAGAE A., LEFEBVRE S., DRETTAKIS G.: Gabor noise by example. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2012) 31*, 4 (July 2012). To appear.

[GSV*14] GILET G., SAUVAGE B., VANHOEY K., DISCHLER J.-M., GHAZANFARPOUR D.: Local random-phase noise for procedural texturing. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 195:1–195:11.

[JBL*06] JANG Y., BOTCHEN R. P., LAUSER A., EBERT D. S., GAITHER K. P., ERTL T.: Enhancing the Interactive Visualization of Procedurally Encoded Multifield Data with Ellipsoidal Basis Functions. *Computer Graphics Forum 25*, 3 (2006), 587–596.

[LD05] LAGAE A., DUTRÉ P.: A procedural object distribution function. *ACM Trans. Graph. 24*, 4 (Oct. 2005), 1442–1461.

[Lew89] LEWIS J. P.: Algorithms for solid noise synthesis. *SIGGRAPH Comput. Graph. 23*, 3 (July 1989), 263–270.

[LLC*10] LAGAE A., LEFEBVRE S., COOK R., DEROSE T., DRETTAKIS G., EBERT D., LEWIS J., PERLIN K., ZWICKER M.: A survey of procedural noise functions. *Computer Graphics Forum 29*, 8 (December 2010), 2579–2600.

[LLDD09] LAGAE A., LEFEBVRE S., DRETTAKIS G., DUTRÉ P.: Procedural noise using sparse gabor convolution. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 54:1–54:10.

[LVLD10] LAGAE A., VANGORP P., LENAERTS T., DUTRÉ P.: Procedural isotropic stochastic textures by example. *Computers '&' Graphics (Special issue on Procedural Methods in Computer Graphics) 34*, 4 (2010), 312–321.

[OL81] OPPENHEIM A. V., LIM J. S.: The Importance of Phase in Signals. *Proceedings of the IEEE 69*, 5 (May 1981), 529–541.

[Per85] PERLIN K.: An image synthesizer. *SIGGRAPH Comput. Graph. 19*, 3 (July 1985), 287–296.

[VGR16] VERSTEEGEN R., GIMEL'FARB G., RIDDLE P.: Texture modelling with nested high-order markov-gibbs random fields. *Computer Vision and Image Understanding 143* (2016), 120 – 134. Inference and Learning of Graphical Models Theory and Applications in Computer Vision and Image Analysis.

[VSLD13] VANHOEY K., SAUVAGE B., LARUE F., DISCHLER J.-M.: On-the-fly multi-scale infinite texturing from example. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 208:1–208:10.

[vW91] VAN WIJK J. J.: Spot noise texture synthesis for data visualization. *SIGGRAPH Comput. Graph. 25*, 4 (July 1991), 309–318.

[WM03] WELSH T., MUELLER K.: A frequency-sensitive point hierarchy for images and volumes. In *Visualization, 2003. VIS 2003. IEEE* (Oct 2003), pp. 425–432.