Color-mapped noise vector fields for generating procedural micro-patterns
Supplemental document

C. Grenier, B. Sauvage, J.-M. Dischler, S. Thery
ICube, Université de Strasbourg, CNRS, France

This supplemental document contains the following elements:

1. A reminder of the background about tiling and blending, and our notations.
2. The first order moment estimation (mean), for a square footprint and then over an arbitrary footprint. This corresponds to Section 5.2 in the paper.
3. The centered second order moment (variance) estimation, for a square footprint and then over an arbitrary footprint. This corresponds to Section 5.3 in the paper.
4. The covariance estimation, for a square footprint and then over an arbitrary footprint. This calculation is used for the normal map filtering in Section 6.5 of the paper.
5. A summary of the main equations (mean, variance and covariance estimators) that are used in the paper and useful for implementation.
6. Results. We show the detailed impact of the estimations. We show other generated patterns. We show the filtering of the procedural phasor noise.

1 Tiling and blending background

We recall here the background about the tiling and blending (T&B) algorithm [HN18, DH19, Bur19]. This algorithm synthesizes a noise N from a discrete input example E by tiling the infinite plane with overlapping hexagonal tiles trimmed in E. Without loss of generality, it is assumed E has zero mean (otherwise both E and N are shifted). Assuming that E is the realization of a Gaussian process, N is evaluated at any location $\mathbf{u}$ as a weighted average of three overlapping tiles :

$$N(\mathbf{u}) = \sum_{i=1}^{3} w_i(\mathbf{u}) E_i(\mathbf{u}) \tag{1}$$

The tiles E_i are trimmed at random locations in E. The weighting functions w_i decrease from 1 at the center of the tile to 0 at the boundary, with $\sum_i w_i^2 = 1$ for all $\mathbf{u}$.

2 First order moment estimation

2.1 MIPmap estimation

In this section we explain how to evaluate a MIPmap of the first order moment of N in real-time, as suggested in [HN18]. A MIPmap of E is pre-computed. We denote $\{E_0, E_1, E_2, \ldots\}$ the MIP hierarchy, with $E_0 = E$ the finest level. A texel at level l has a square footprint $\mathbb{P}$ with center $\dot{\mathbb{P}}$.

$$E_l(\dot{\mathbb{P}}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u} \in \mathbb{P}} E(\mathbf{u}) \tag{2}$$

Now we approximate the mean of N over $\mathbb{P}$ by

$$
\begin{aligned}
\overline{N}(\mathbb{P}) &\stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} N(\mathbf{u}) && (3)\\
&\approx \sum_{i=1}^{3} w_i(\dot{\mathbb{P}}) E_{l,i}(\mathbb{P}) \stackrel{\text{def}}{=} \widehat{N}_l(\mathbb{P}) && (4)
\end{aligned}
$$

where the tile $E_{l,i}$ has the same shape as E_i but is trimmed in E_l. The approximation is due to $w_i(\mathbf{u}) \approx w_i(\dot{\mathbb{P}})$ for all $\mathbf{u} \in \mathbb{P}$, which is considered constant over the footprint and evaluated at its center.

2.2 Estimation over a arbitrary footprint

At this stage, we are able to estimate the average of N over a square footprint $\mathbb{P}$ corresponding to a texel in the MIPmap. We now want to estimate the value over an arbitrary footprint $\mathcal{P}$ represented by its compactly supported weighting function $w_\mathcal{P}$. To achieve this in real-time, we leverage the previous MIPmap and a discretization $w_\mathcal{P}(\mathbb{P})$ over the texels of the MIPmaps. We assume it to be normalized, i.e.

$$\sum_{\mathbb{P}\in\mathcal{P}} w_\mathcal{P}(\mathbb{P}) = 1 \qquad (5)$$

The basic idea is to: cut the integral into pieces $\mathbb{P} \in \mathcal{P}$; approximate $w_\mathcal{P}$ constant over $\mathbb{P}$; use the previous MIPmap estimation.

$$
\begin{aligned}
\overline{N}(\mathcal{P}) &\stackrel{\text{def}}{=} \int_\mathcal{P} w_\mathcal{P}(\mathbf{u}) N(\mathbf{u}) \mathrm{d}\mathbf{u} && (6)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} \int_\mathbb{P} w_\mathcal{P}(\mathbf{u}) N(\mathbf{u}) \mathrm{d}\mathbf{u} && (7)\\
&\approx \sum_{\mathbb{P}\in\mathcal{P}} w_\mathcal{P}(\mathbb{P}) \int_\mathbb{P} N(\mathbf{u}) \mathrm{d}\mathbf{u} && (8)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} w_\mathcal{P}(\mathbb{P}) \overline{N}(\mathbb{P}) && (9)\\
&\approx \sum_{\mathbb{P}\in\mathcal{P}} w_\mathcal{P}(\mathbb{P}) \widehat{N_l}(\mathbb{P}) \stackrel{\text{def}}{=} \widehat{N}_l(\mathcal{P}) && (10)
\end{aligned}
$$

3 Centered second order moment estimation

3.1 MIPmap estimation

We now want to approximate a MIPmap of the second order moment (variance) of N in real-time.

$$\sigma^2(\mathbb{P}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(N(\mathbf{u}) - \overline{N}(\mathbb{P})\right)^2 \tag{11}$$

$$= \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\mathbf{u}) E_i(\mathbf{u}) - \overline{N}(\mathbb{P})\right)^2 \tag{12}$$

$$\approx \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\mathbf{u}) E_i(\mathbf{u}) - \widehat{N}_l(\mathbb{P})\right)^2 \tag{13}$$

$$= \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\mathbf{u}) E_i(\mathbf{u}) - w_i(\dot{\mathbb{P}}) E_{l,i}(\mathbb{P})\right)^2 \tag{14}$$

$$\approx \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\dot{\mathbb{P}}) \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\right)^2 \tag{15}$$

Here we can develop the inner sum:

$$\left(\sum_{i=1}^{3} w_i(\dot{\mathbb{P}}) \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\right)^2 = \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)^2 \tag{16}$$

$$+ 2 \sum_{1\leq i<j\leq} w_i(\dot{\mathbb{P}}) w_j(\dot{\mathbb{P}}) \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right) \left(E_j(\mathbf{u}) - E_{l,j}(\mathbb{P})\right) \tag{17}$$

$$\approx \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)^2 \tag{18}$$

by neglecting the cross-products. Note that it is possible to neglect the cross-products because we compute the *centered* moment (σ^2); this would not be possible for the *non-centered* moment (N^2) because the cross-products contribute a lot. Then we plug the approximation in the formula of σ^2:

$$\sigma^2(\mathbb{P}) \approx \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)^2 \tag{19}$$

$$\approx \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)^2 \tag{20}$$

$$= \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) V_{l,i}(\mathbb{P}) \stackrel{\text{def}}{=} \widehat{\sigma^2}_l(\mathbb{P}) \tag{21}$$

where $V_{l,i}$ is a tile trimmed in V_l. We define $\{V_0, V_1, V_2, \ldots\}$ as a MIPmap of variance:

$$V_l(\mathbb{P}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(E(\mathbf{u}) - E_l(\mathbb{P})\right)^2. \tag{22}$$

3.2 Estimation over an arbitrary footprint

Here we estimate the variance $\sigma^2(\mathcal{P})$ over an arbitrary footprint $\mathcal{P}$ represented by its compactly supported weighting function $w_{\mathcal{P}}$. To achieve this in real-time, we leverage the previous MIPmaps and a discretization $w_{\mathcal{P}}(\mathbb{P})$ over the texels of the MIPmaps. Again, we assume it to be normalized, i.e.

$$\sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) = 1 \tag{23}$$

We can not follow exactly the same derivation as the first order, because the variance over a union $\mathcal{P} = \bigcup \mathbb{P}$ is not the average of variances. Instead, we write

$$\sigma^2(\mathcal{P}) = \overline{N^2}(\mathcal{P}) - \left(\overline{N}(\mathcal{P})\right)^2 \tag{24}$$

We approximate $\overline{N}(\mathcal{P}) \approx \widehat{N}_l(\mathcal{P})$ defined in Section 2. We are left with the problem of approximating the *non-centered* second order moment

$$\overline{N^2}(\mathcal{P}) \stackrel{\text{def}}{=} \int_{\mathcal{P}} w_{\mathcal{P}}(\mathbf{u}) N^2(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{25}$$

which can be averaged over a union of square footprints:

$$\begin{aligned}
\overline{N^2}(\mathcal{P}) &= \sum_{\mathbb{P}\in\mathcal{P}} \int_{\mathbb{P}} w_{\mathcal{P}}(\mathbf{u}) N^2(\mathbf{u}) \mathrm{d}\mathbf{u} && (26)\\
&\approx \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \int_{\mathbb{P}} N^2(\mathbf{u}) \mathrm{d}\mathbf{u} && (27)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \overline{N^2}(\mathbb{P}) && (28)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \left(\overline{N}(\mathbb{P})^2 + \sigma^2(\mathbb{P})\right) && (29)\\
&\approx \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \left(\widehat{N}_l(\mathbb{P})^2 + \widehat{\sigma^2}_l(\mathbb{P})\right) \stackrel{\text{def}}{=} \widehat{N^2}_l(\mathcal{P}) && (30)
\end{aligned}$$

4 Covariance estimation

For the purpose of normal map filtering, we investigate here the estimation of the covariance of two noises N and N', which correspond in the paper to the slopes in x and y.

We assume the noises to be generated using the same T&B from two inputs E and E'. It is *very important* to note that we use the same random numbers, so that the tiles (E_i and E'_i in the equations below) are at the same positions in E and in E'. Otherwise the MIPmap would not be coherent.

To do so, we adapt the equations of the previous section.

4.1 MIPmap estimation

We want to approximate

$$
\begin{aligned}
cov(\mathbb{P}) &\stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(N(\mathbf{u}) - \overline{N}(\mathbb{P})\right)\left(N'(\mathbf{u}) - \overline{N'}(\mathbb{P})\right) && (31)\\
&= \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\mathbf{u})E_i(\mathbf{u}) - \overline{N}(\mathbb{P})\right)\left(\sum_{i=1}^{3} w_i(\mathbf{u})E'_i(\mathbf{u}) - \overline{N'}(\mathbb{P})\right) && (32)\\
&\approx \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\mathbf{u})E_i(\mathbf{u}) - \widehat{N}_l(\mathbb{P})\right)\left(\sum_{i=1}^{3} w_i(\mathbf{u})E'_i(\mathbf{u}) - \widehat{N'}_l(\mathbb{P})\right) && (33)\\
&= \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\mathbf{u})E_i(\mathbf{u}) - w_i(\dot{\mathbb{P}})E_{l,i}(\mathbb{P})\right)\left(\sum_{i=1}^{3} w_i(\mathbf{u})E'_i(\mathbf{u}) - w_i(\dot{\mathbb{P}})E'_{l,i}(\mathbb{P})\right) && (34)\\
&\approx \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(\sum_{i=1}^{3} w_i(\dot{\mathbb{P}})\left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\right)\left(\sum_{i=1}^{3} w_i(\dot{\mathbb{P}})\left(E'_i(\mathbf{u}) - E'_{l,i}(\mathbb{P})\right)\right) && (35)
\end{aligned}
$$

Here we can develop the product of inner sums:

$$
\begin{aligned}
\left(\ldots\right)\left(\ldots'\right) &= \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}})\left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\left(E'_i(\mathbf{u}) - E'_{l,i}(\mathbb{P})\right) && (36)\\
&\quad + \sum_{i\neq j} w_i(\dot{\mathbb{P}})w_j(\dot{\mathbb{P}})\left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\left(E'_j(\mathbf{u}) - E'_{l,j}(\mathbb{P})\right) && (37)\\
&\approx \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}})\left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\left(E'_i(\mathbf{u}) - E'_{l,i}(\mathbb{P})\right) && (38)
\end{aligned}
$$

by neglecting the cross-products. Then we plug the approximation in the formula of *cov*:

$$
\begin{aligned}
cov(\mathbb{P}) &\approx \frac{1}{\#\mathbb{P}} \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}})\left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\left(E'_i(\mathbf{u}) - E'_{l,i}(\mathbb{P})\right) && (39)\\
&\approx \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(E_i(\mathbf{u}) - E_{l,i}(\mathbb{P})\right)\left(E'_i(\mathbf{u}) - E'_{l,i}(\mathbb{P})\right) && (40)\\
&= \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}})C_{l,i}(\mathbb{P}) \stackrel{\text{def}}{=} \widehat{cov}_l(\mathbb{P}) && (41)
\end{aligned}
$$

where $C_{l,i}$ is a tile trimmed in C_l. We define $\{C_0, C_1, C_2, \ldots\}$ as a MIPmap of covariance:

$$
C_l(\mathbb{P}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(E(\mathbf{u}) - E_l(\mathbb{P})\right)\left(E'(\mathbf{u}) - E'_l(\mathbb{P})\right). \qquad (42)
$$

4.2 Estimation over a arbitrary footprint

Here we estimate the covariance $cov(\mathcal{P})$. As for the variance, the covariance over a union $\mathcal{P} = \bigcup \mathbb{P}$ is not the average of covariances. Instead, we write

$$cov(\mathcal{P}) = \overline{NN'}(\mathcal{P}) - \overline{N}(\mathcal{P})\overline{N'}(\mathcal{P}) \tag{43}$$

and we are left with the problem of approximating

$$\begin{aligned}
\overline{NN'}(\mathcal{P}) &\overset{\text{def}}{=} \int_{\mathcal{P}} w_{\mathcal{P}}(\mathbf{u})N(\mathbf{u})N'(\mathbf{u})\mathrm{d}\mathbf{u} && (44)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} \int_{\mathbb{P}} w_{\mathcal{P}}(\mathbf{u})N(\mathbf{u})N'(\mathbf{u})\mathrm{d}\mathbf{u} && (45)\\
&\approx \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \int_{\mathbb{P}} N(\mathbf{u})N'(\mathbf{u})\mathrm{d}\mathbf{u} && (46)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P})\overline{NN'}(\mathbb{P}) && (47)\\
&= \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \left(\overline{N}(\mathbb{P})\overline{N'}(\mathbb{P}) + cov(\mathbb{P})\right) && (48)\\
&\approx \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \left(\widehat{N}_l(\mathbb{P})\widehat{N'}_l(\mathbb{P}) + \widehat{cov}_l(\mathbb{P})\right) \overset{\text{def}}{=} \widehat{NN'}_l(\mathcal{P}) && (49)
\end{aligned}$$

5 Summary

5.1 Mean estimation

The input is E_0. A MIP hierarchies is precomputed. $\{E_0, E_1, E_2, \ldots\}$ is the standard MIP hierarchy. At level l, we denote a texel or its square footprint as $\mathbb{P}$, with center $\dot{\mathbb{P}}$.

$$E_l(\mathbb{P}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} E_0(\mathbf{u}) \tag{50}$$

The mean over a texel is estimated by

$$\widehat{N}_l(\mathbb{P}) \stackrel{\text{def}}{=} \sum_{i=1}^{3} w_i(\dot{\mathbb{P}}) E_{l,i}(\mathbb{P}) \approx \overline{N}(\mathbb{P}) \tag{51}$$

$\mathcal{P} = \bigcup \mathbb{P}$ is a footprint which covers several texels $\mathbb{P}$ at level l. The texels are weighted by $w_{\mathcal{P}}(\mathbb{P}) > 0$ that sum up to 1. The mean over is estimated by

$$\widehat{N}_l(\mathcal{P}) \stackrel{\text{def}}{=} \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \widehat{N}_l(\mathbb{P}) \approx \overline{N}(\mathcal{P}) \tag{52}$$

5.2 Variance estimation

The input is E_0. Two MIP hierarchies are precomputed. $\{E_0, E_1, E_2, \ldots\}$ is the standard MIP hierarchy. We define a MIPmap of variance $\{V_0, V_1, V_2, \ldots\}$. At level l, we denote a texel or its square footprint as $\mathbb{P}$, with center $\dot{\mathbb{P}}$.

$$V_l(\mathbb{P}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(E(\mathbf{u}) - E_l(\mathbb{P})\right)^2. \tag{53}$$

The variance over a texel is estimated by

$$\widehat{\sigma^2}_l(\mathbb{P}) \stackrel{\text{def}}{=} \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) V_{l,i}(\mathbb{P}) \approx \sigma^2(\mathbb{P}) \tag{54}$$

$\mathcal{P} = \bigcup \mathbb{P}$ is a footprint which covers several texels $\mathbb{P}$ at level l. The texels are weighted by $w_{\mathcal{P}}(\mathbb{P}) > 0$ that sum up to 1. The variance over is estimated by

$$\widehat{\sigma^2}_l(\mathcal{P}) \stackrel{\text{def}}{=} \widehat{N^2}_l(\mathcal{P}) - \left(\widehat{N}_l(\mathcal{P})\right)^2 \approx \sigma^2(\mathcal{P}) \tag{55}$$

with

$$\widehat{N^2}_l(\mathcal{P}) \stackrel{\text{def}}{=} \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \left(\widehat{N}_l(\mathbb{P})^2 + \widehat{\sigma^2}_l(\mathbb{P})\right) \approx \overline{N^2}(\mathcal{P}) \tag{56}$$

5.3 Covariance estimation

We define a MIPmap of covariance $\{C_0, C_1, C_2, \ldots\}$ as

$$C_l(\mathbb{P}) \stackrel{\text{def}}{=} \frac{1}{\#\mathbb{P}} \sum_{\mathbf{u}\in\mathbb{P}} \left(E(\mathbf{u}) - E_l(\mathbb{P})\right)\left(E'(\mathbf{u}) - E'_l(\mathbb{P})\right). \tag{57}$$

The covariance over a texel are estimated by

$$\widehat{cov}_l(\mathbb{P}) \stackrel{\text{def}}{=} \sum_{i=1}^{3} w_i^2(\dot{\mathbb{P}}) C_{l,i}(\mathbb{P}) \approx cov(\mathbb{P}) \tag{58}$$

The covariance over $\mathcal{P} = \bigcup \mathbb{P}$ is estimated by

$$\widehat{NN'}_l(\mathcal{P}) \stackrel{\text{def}}{=} \sum_{\mathbb{P}\in\mathcal{P}} w_{\mathcal{P}}(\mathbb{P}) \left(\widehat{N}_l(\mathbb{P})\widehat{N'}_l(\mathbb{P}) + \widehat{cov}_l(\mathbb{P}) \right) \approx \overline{NN'}(\mathcal{P}) \tag{59}$$

$$\widehat{cov}_l(\mathcal{P}) \stackrel{\text{def}}{=} \widehat{NN'}_l(\mathcal{P}) - \widehat{N}_l(\mathcal{P})\widehat{N'}_l(\mathcal{P}) \approx cov(\mathcal{P}) \tag{60}$$

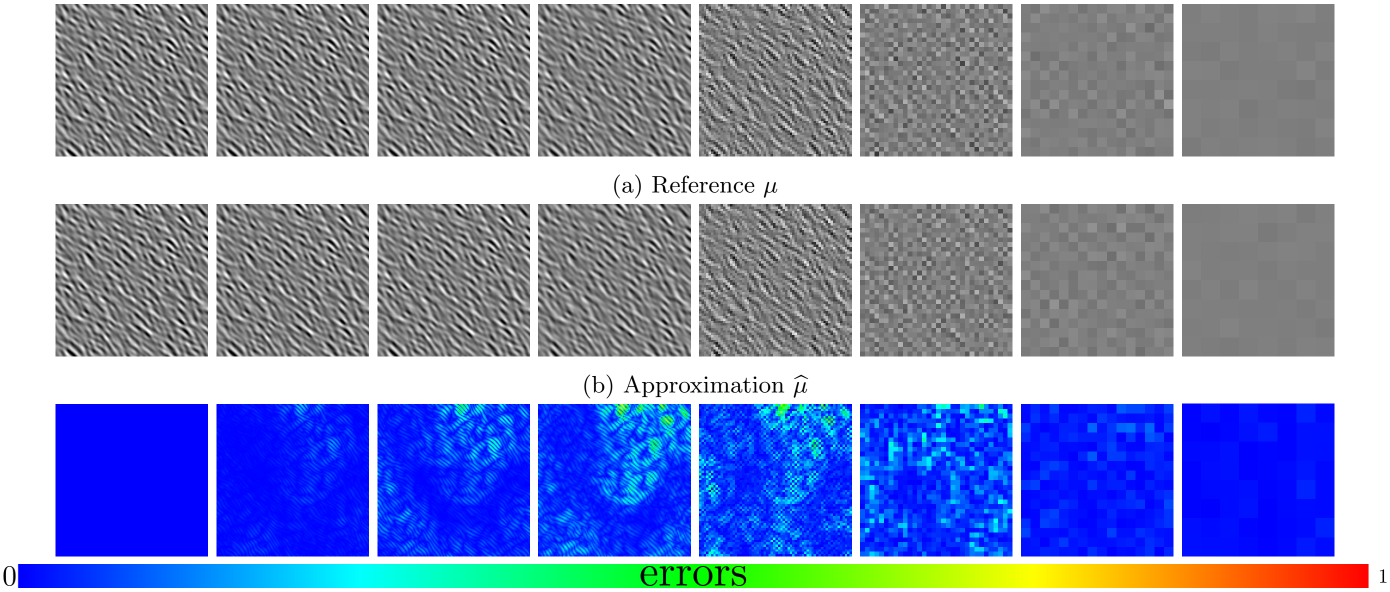
6 Results

In this section:

1. We present in detail the impact of the approximation of the mean and variance.
2. We examine the impact of these approximations on the filtering of the patterns.
3. We show other generated patterns, using various color-map and input noises.
4. We examine the filtering of the procedural phasor noise.

6.1 First order moment and standard deviation estimation

In this section, we measure the errors due to mean and variance estimation, compared to the reference. We also plot the standard deviation, as it is commensurable to the mean –while the the variance is not.

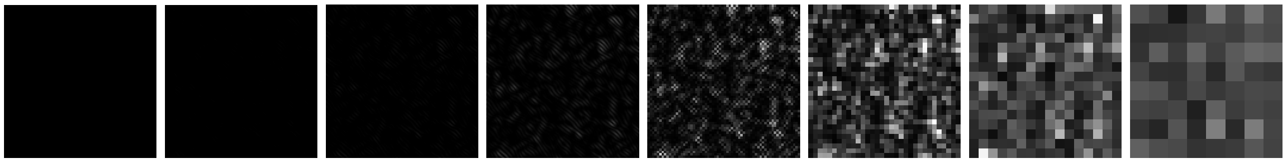


(a) Reference μ

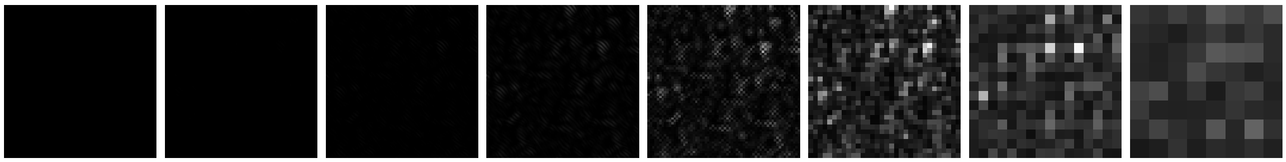
(b) Approximation $\widehat{\mu}$

(c) Error $||\widehat{\mu} - \mu||$

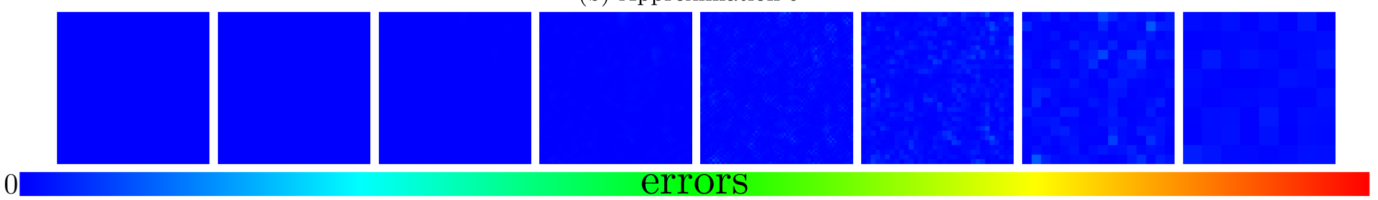
Figure 1: Approximation $\widehat{\mu}$ of the first order moment μ. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference σ^2

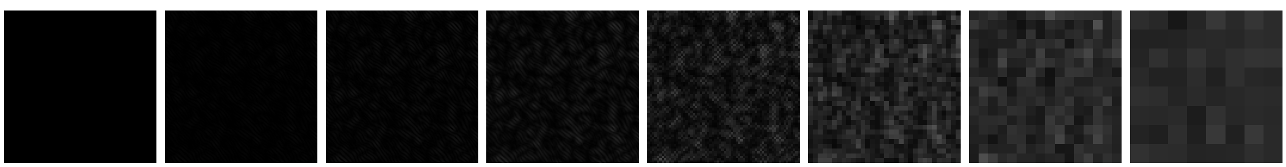


(b) Approximation $\widehat{\sigma^2}$

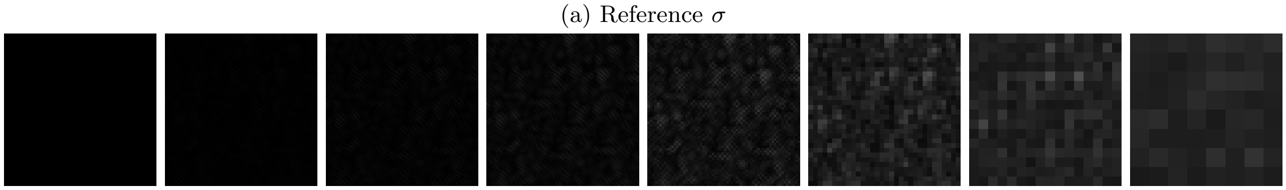


(c) Error $||\widehat{\sigma^2} - \sigma^2||$

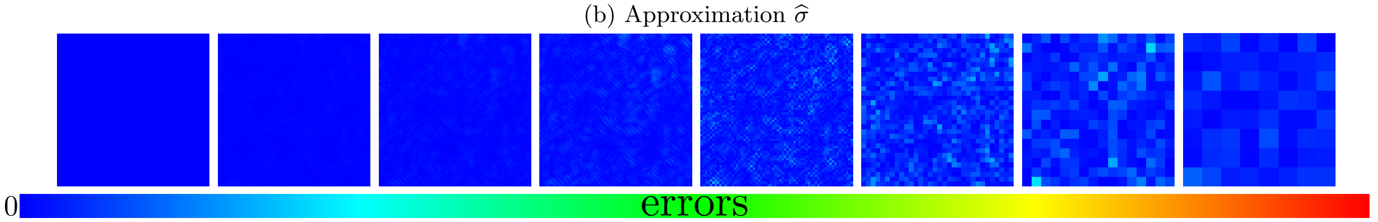
Figure 2: Approximation $\widehat{\sigma^2}$ of the variance σ^2 scaled by a factor 10 for the visibility. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference σ



(b) Approximation $\widehat{\sigma}$



(c) Error $||\widehat{\sigma} - \sigma||$

Figure 3: Approximation $\widehat{\sigma}$ of the standard deviation σ. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.

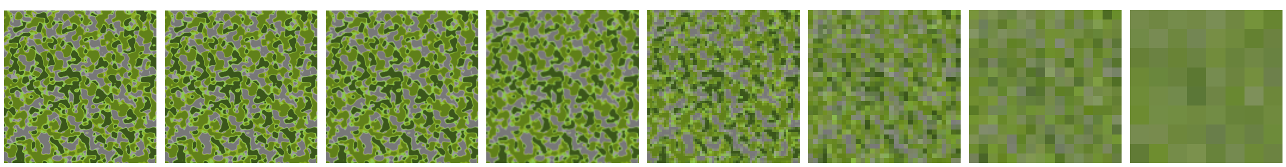
6.2 Approximation for colored noise filtering

We compare errors due to the different approximations $\widehat{S}$, $\widehat{\mu}$ and $\widehat{\sigma}$ separately:

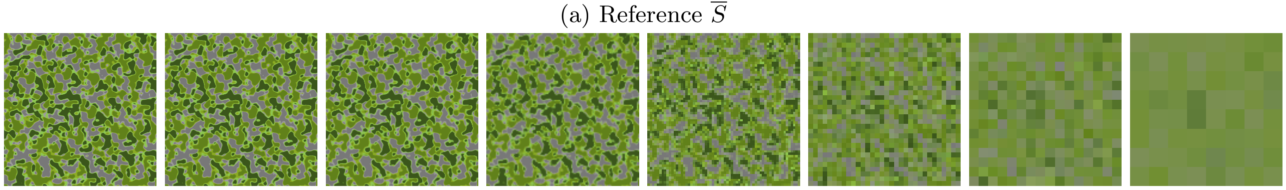
1. $\widehat{S}(\mu, \sigma)$, using the exact values of μ and σ (Figure 4 & 8).
2. $\widehat{S}(\widehat{\mu}, \sigma)$, using the estimated mean value $\widehat{\mu}$ and the exact value of σ (Figure 5 & 9).
3. $\widehat{S}(\mu, \widehat{\sigma})$, using the exact value of μ and the estimated standard deviation value $\widehat{\sigma}$ (Figure 6 & 10).
4. $\widehat{S}(\widehat{\mu}, \widehat{\sigma})$, using both the estimated mean $\widehat{\mu}$ and standard deviation $\widehat{\sigma}$ (Figure 7 & 11).

We present the previous approximation for two different color-map. The first one is composed of different shades of green and the second one has saturated red, green, and blue colors. Those two color-maps bring out the impact of the discontinuities in the color-map used. Thus, errors are located in the same area but are more accentuated if the color-map presents strong color changes than smooth ones.

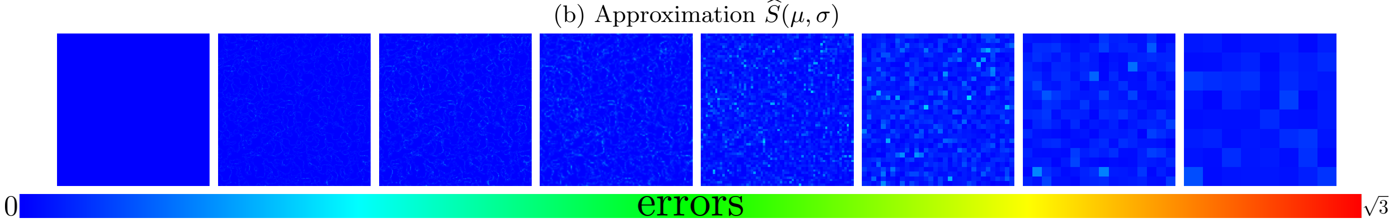
6.2.1 Green color-map with smooth color changes



(a) Reference $\overline{S}$

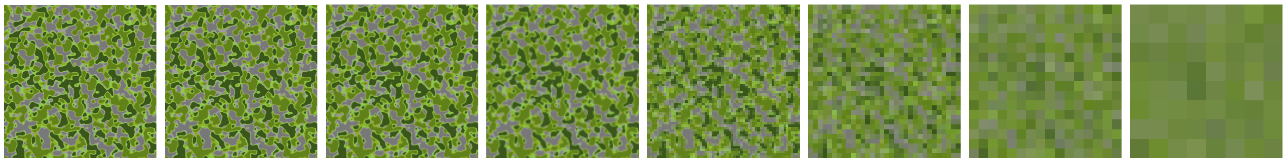


(b) Approximation $\widehat{S}(\mu, \sigma)$

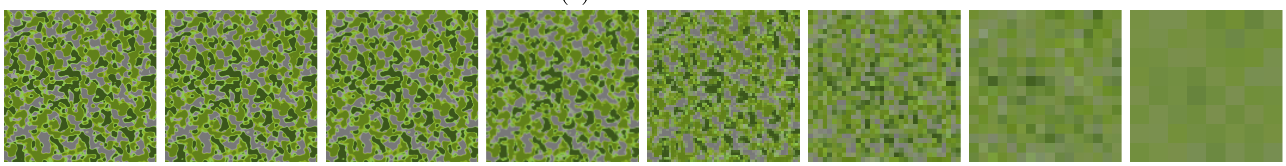


(c) Error $||\widehat{S} - \overline{S}||$

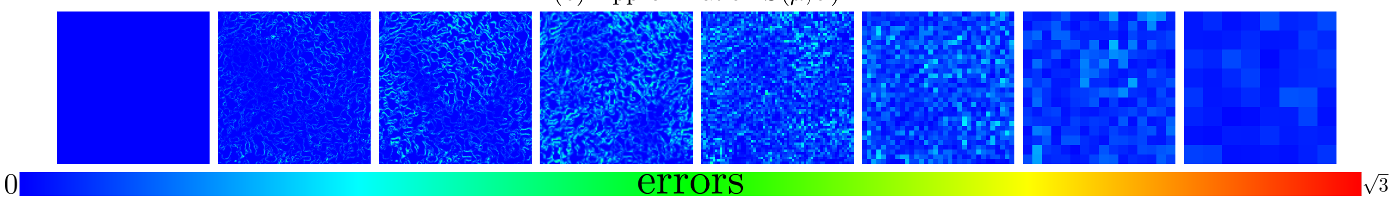
Figure 4: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with exact computation of the mean and the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$

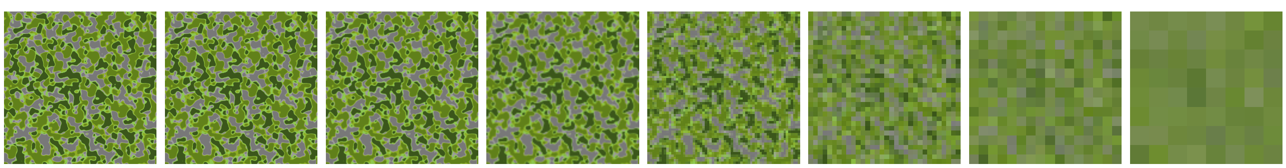


(b) Approximation $\widehat{S}(\widehat{\mu}, \sigma)$

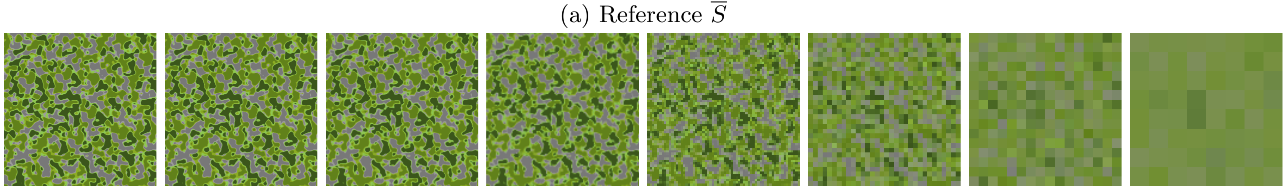


(c) Error $||\widehat{S} - \overline{S}||$

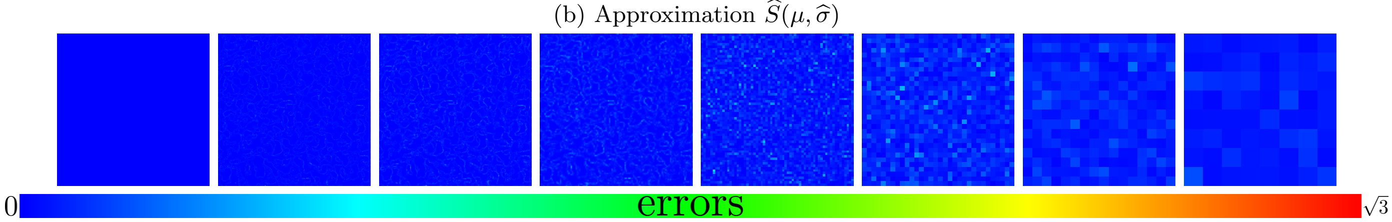
Figure 5: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with approximation of the mean and exact computation of the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$

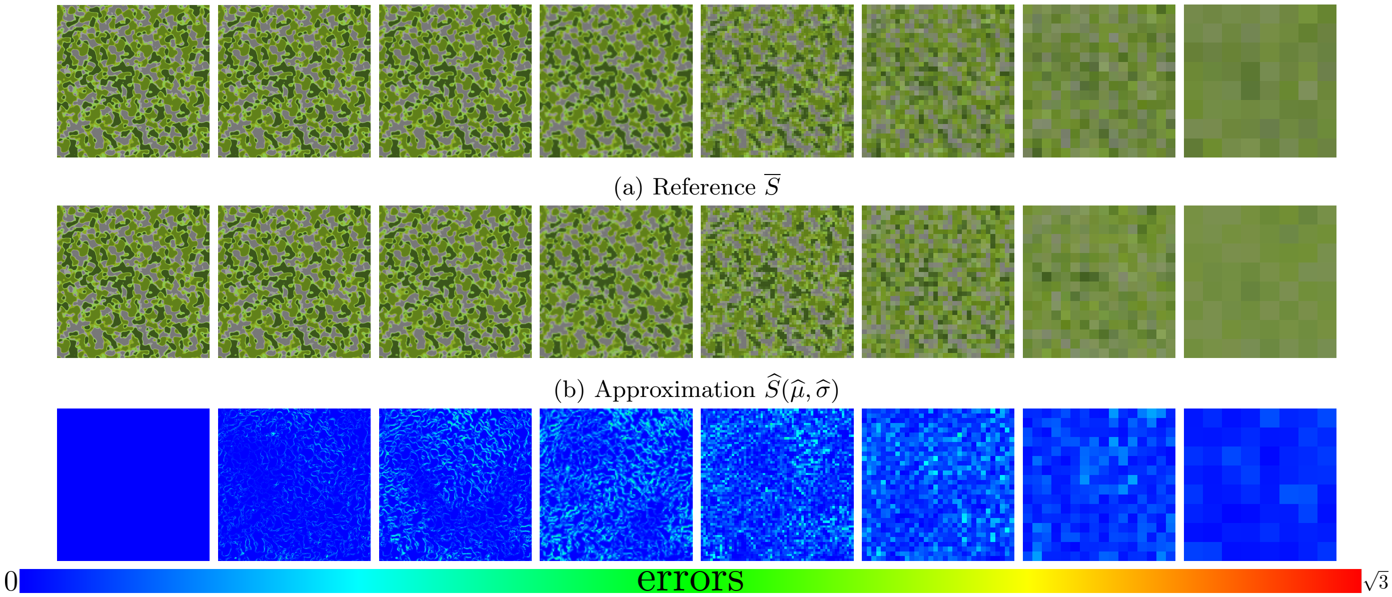


(b) Approximation $\widehat{S}(\mu, \widehat{\sigma})$



(c) Error $||\widehat{S} - \overline{S}||$

Figure 6: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with exact computation of the mean and approximation of the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



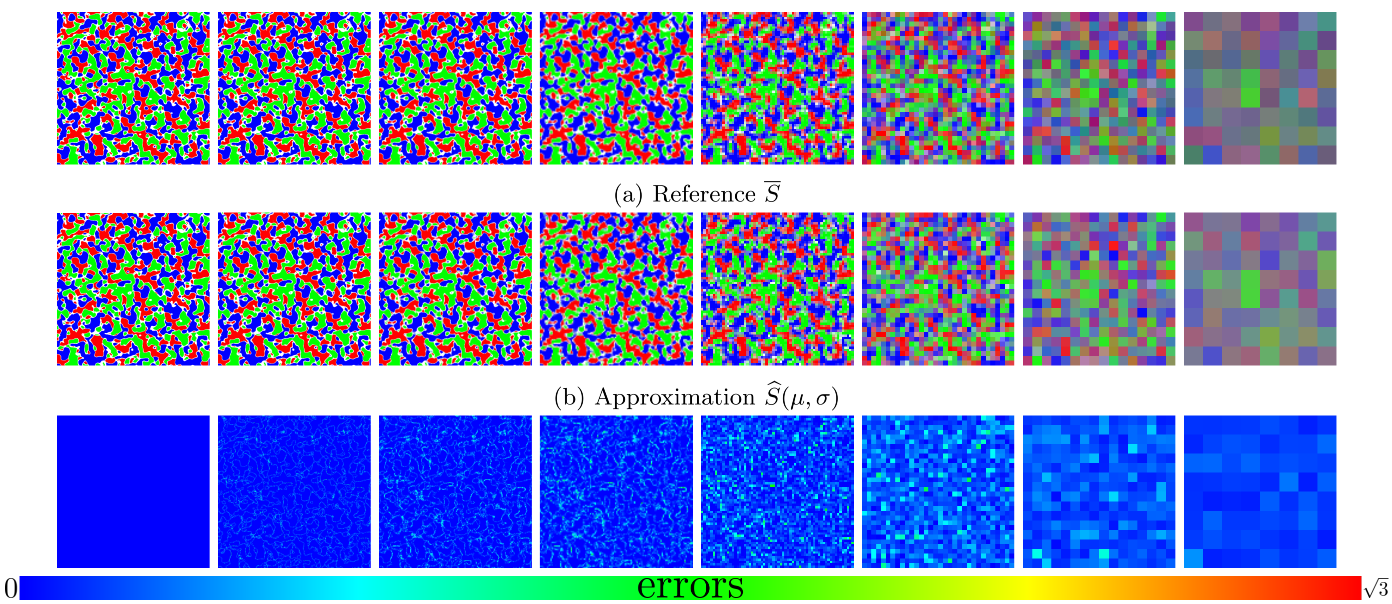
(a) Reference $\overline{S}$

(b) Approximation $\widehat{S}(\widehat{\mu}, \widehat{\sigma})$

(c) Error $||\widehat{S} - \overline{S}||$

Figure 7: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with approximation of the mean and the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.

6.2.2 RGB color-map with strong color changes

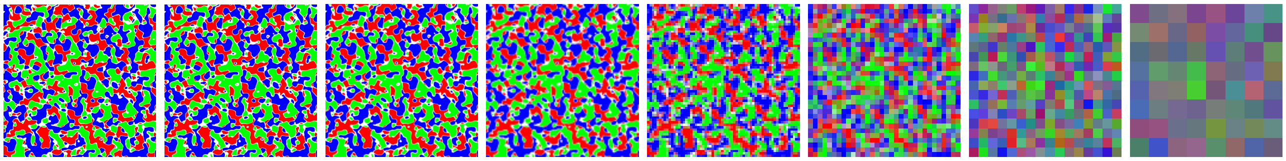


(a) Reference $\overline{S}$

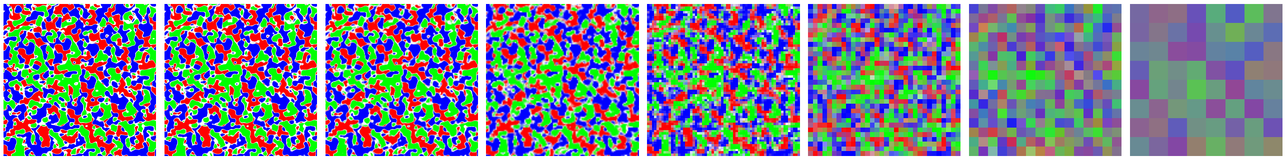
(b) Approximation $\widehat{S}(\mu, \sigma)$

(c) Error $||\widehat{S} - \overline{S}||$

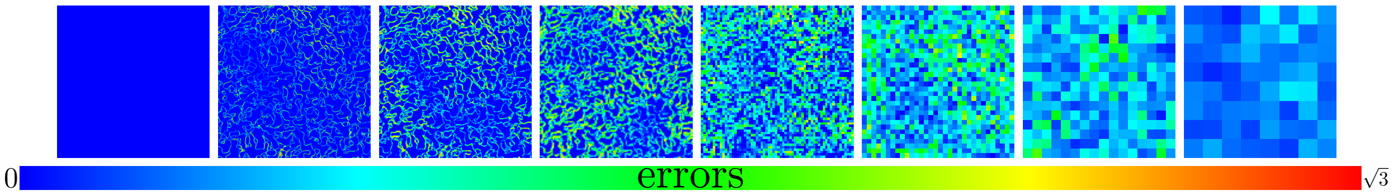
Figure 8: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with exact computation of the mean and the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$

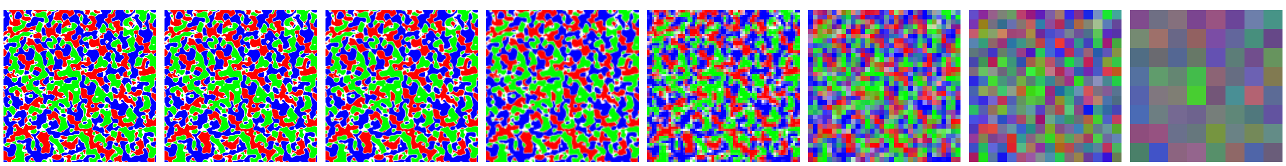


(b) Approximation $\widehat{S}(\widehat{\mu}, \sigma)$

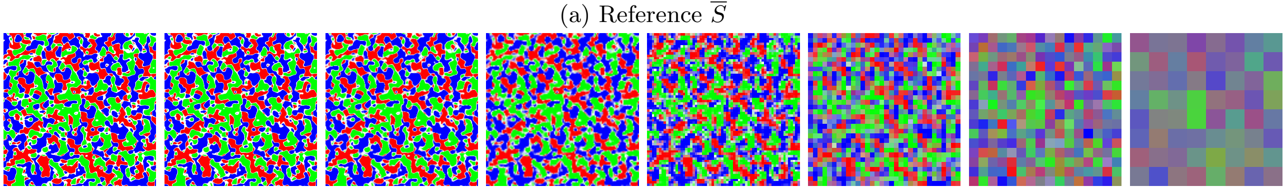


(c) Error $||\widehat{S} - \overline{S}||$

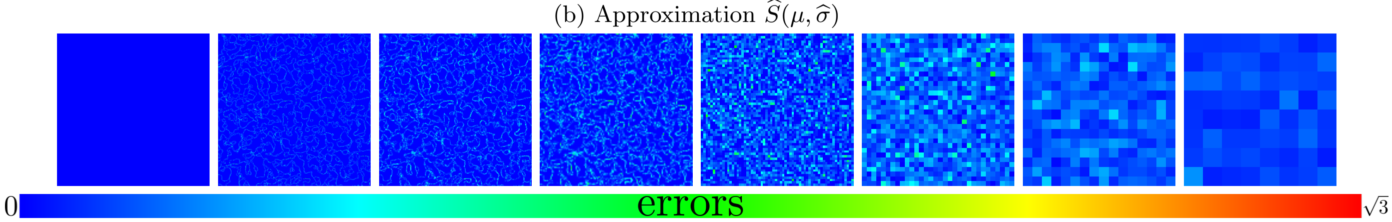
Figure 9: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with approximation of the mean and exact computation of the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$

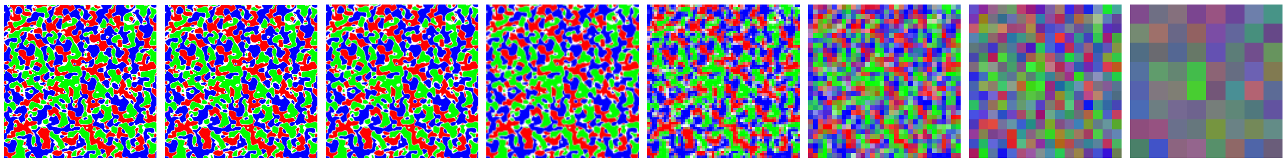


(b) Approximation $\widehat{S}(\mu, \widehat{\sigma})$

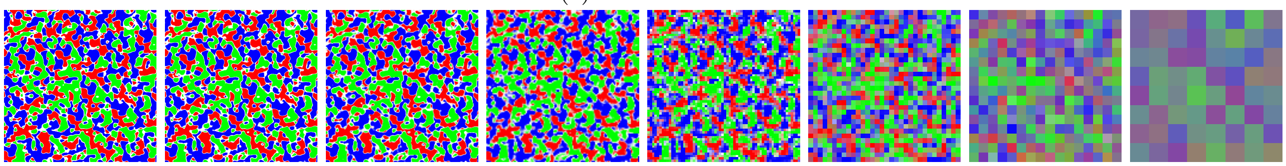


(c) Error $||\widehat{S} - \overline{S}||$

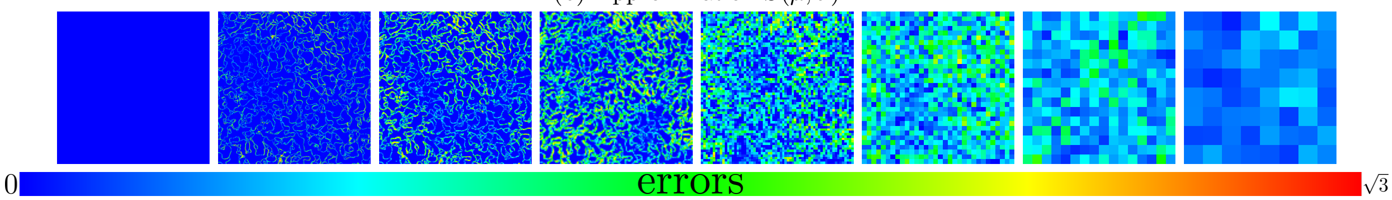
Figure 10: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with exact computation of the mean and approximation of the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$



(b) Approximation $\widehat{S}(\widehat{\mu}, \widehat{\sigma})$



(c) Error $||\widehat{S} - \overline{S}||$

Figure 11: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with approximation of the mean and the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.

6.3 Several examples of generated patterns

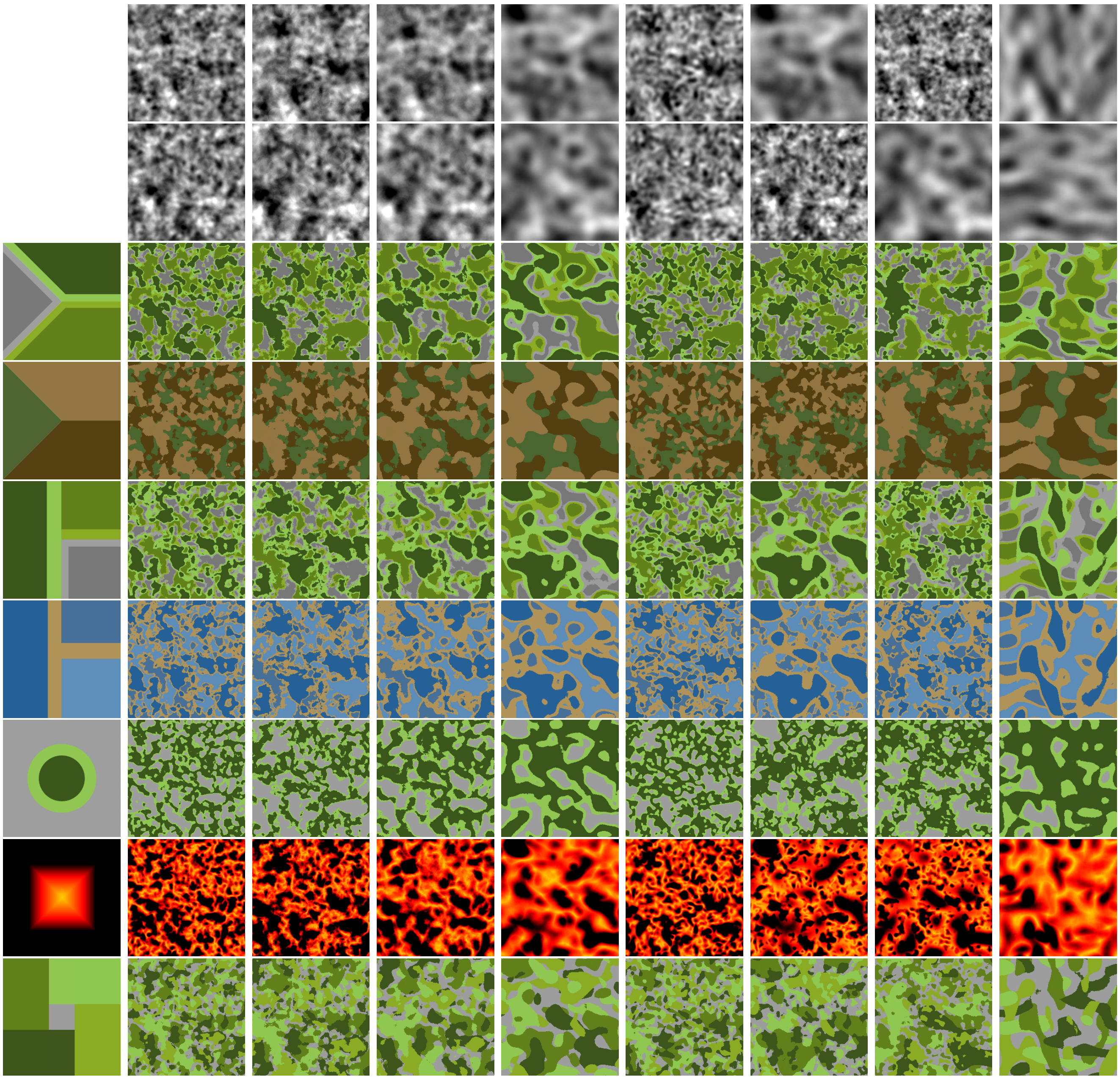


Figure 12: Examples of generated patterns. The first and second rows are the input noises and the left column is the color-map.

6.4 Filtering of the procedural phasor noise

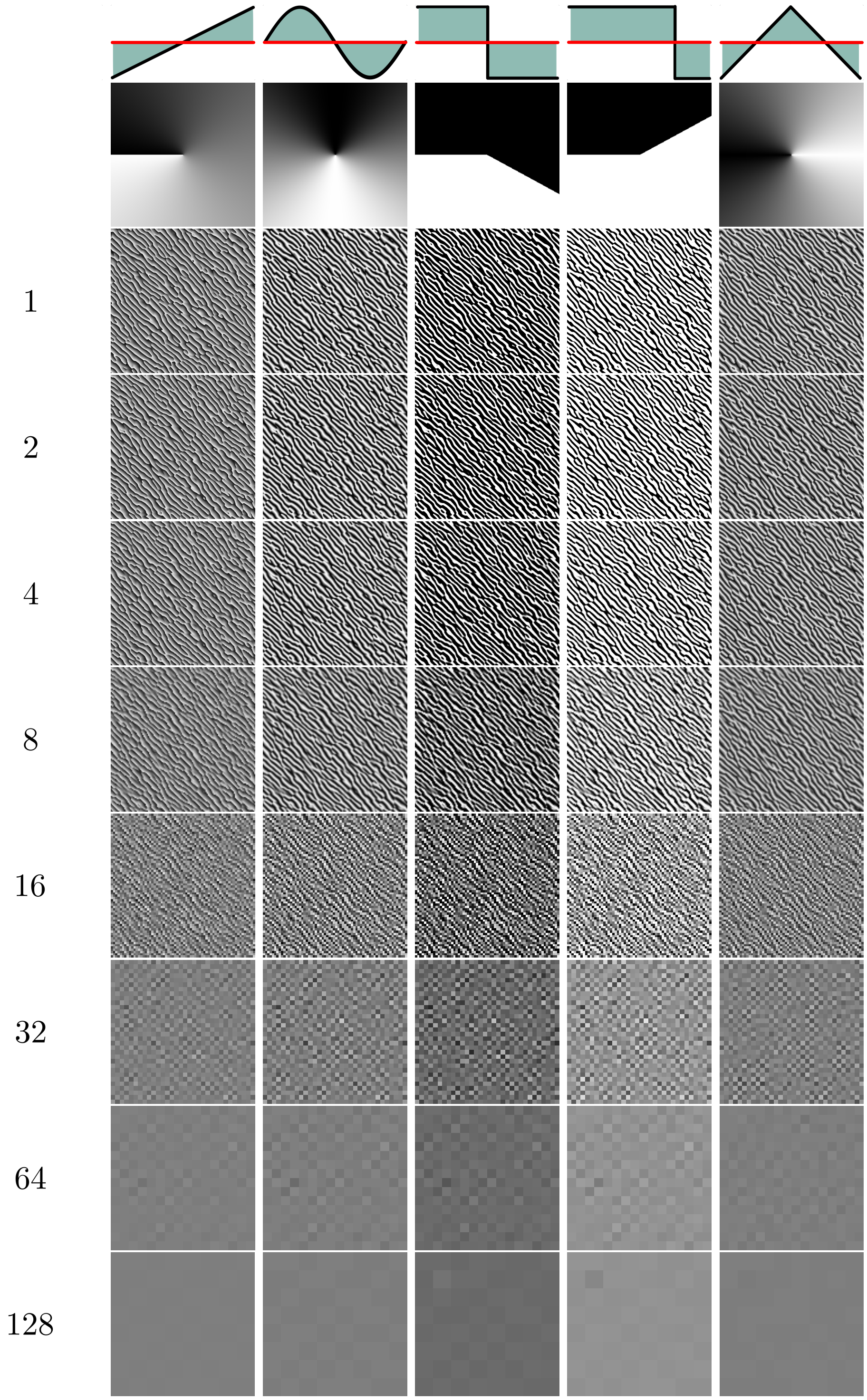
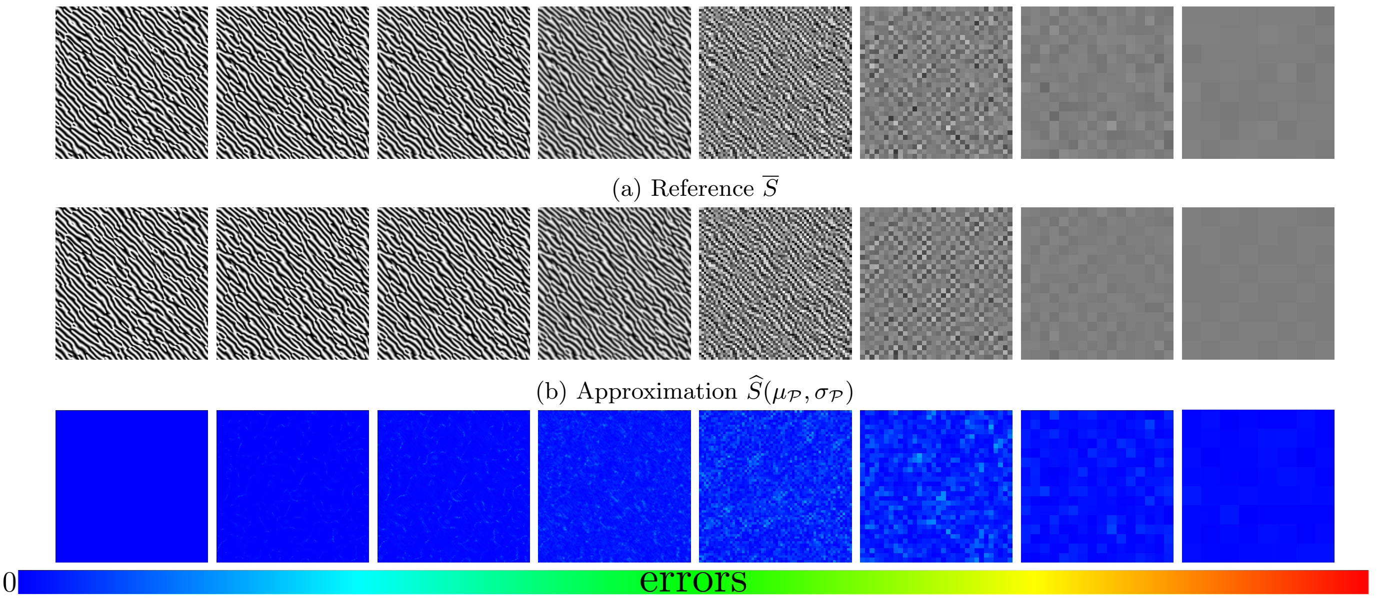


Figure 13: Result of the filtering of the procedural phasor noise with different profile function. Top rows: profiles, and color-maps $profile \circ atan2$. Then, from top to bottom: approximation $\widehat{S}(\widehat{\mu_{\mathcal{P}}}, \widehat{\sigma_{\mathcal{P}}})$ for footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.

We compare errors due to the different approximations $\widehat{S}$, $\widehat{\mu}$ and $\widehat{\sigma}$ separately:

1. $\widehat{S}(\mu, \sigma)$, using the exact values of μ and σ (Figure 14);
2. $\widehat{S}(\widehat{\mu}, \sigma)$, using the estimated mean value $\widehat{\mu}$ and the exact value of σ (Figure 15);
3. $\widehat{S}(\mu, \widehat{\sigma})$, using the exact value of μ and the estimated standard deviation value $\widehat{\sigma}$ (Figure 16);
4. $\widehat{S}(\widehat{\mu}, \widehat{\sigma})$, using both the estimated mean $\widehat{\mu}$ and standard deviation $\widehat{\sigma}$ (Figure 17);

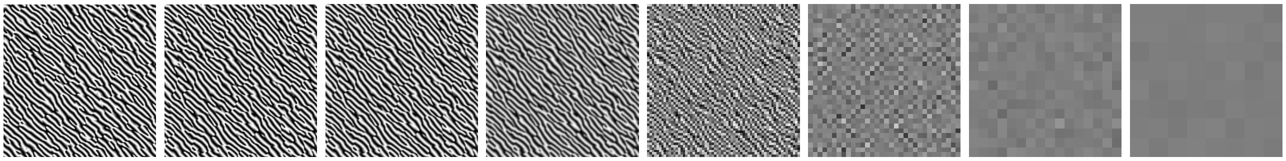


(a) Reference $\overline{S}$

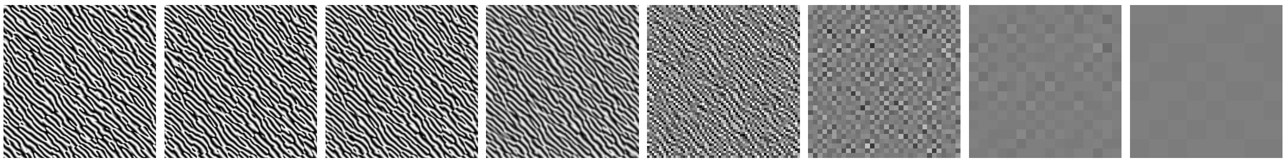
(b) Approximation $\widehat{S}(\mu_{\mathcal{P}}, \sigma_{\mathcal{P}})$

(c) Error $||\widehat{S} - \overline{S}||$

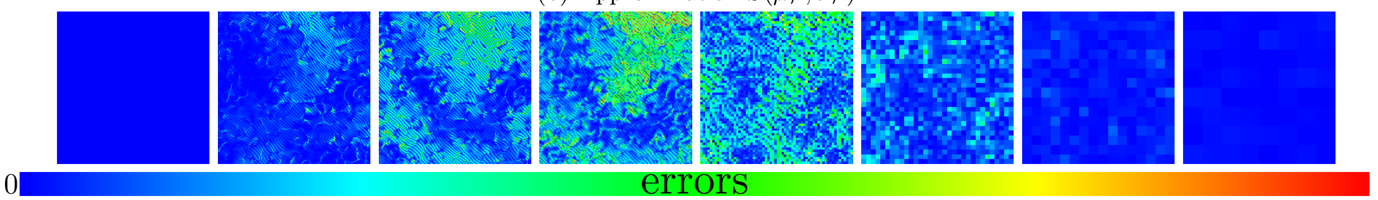
Figure 14: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with exact computation of the mean and the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$

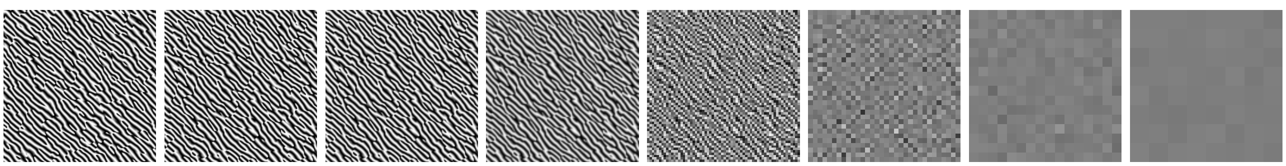


(b) Approximation $\widehat{S}(\widehat{\mu_{\mathcal{P}}}, \sigma_{\mathcal{P}})$

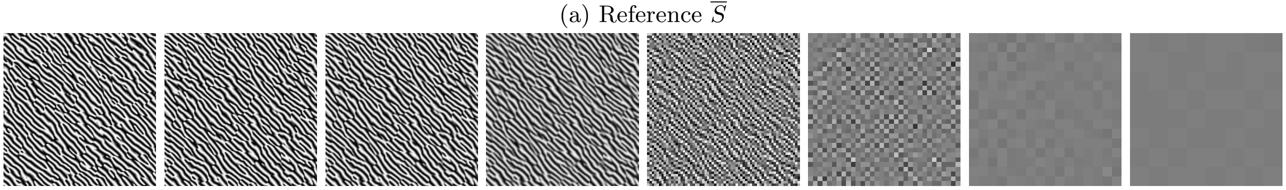


(c) Error $||\widehat{S} - \overline{S}||$

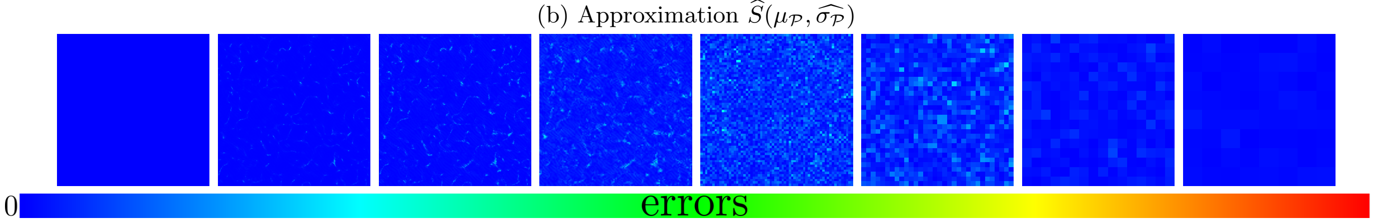
Figure 15: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with approximation of the mean and exact computation of the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$

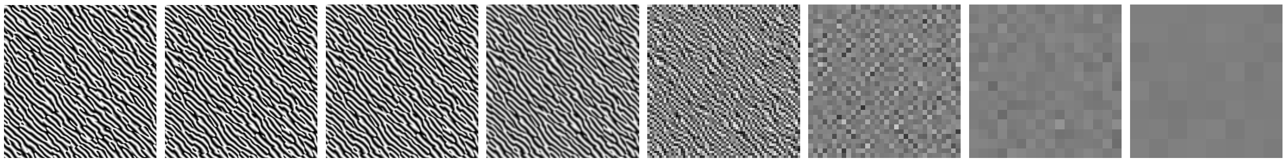


(b) Approximation $\widehat{S}(\mu_{\mathcal{P}}, \widehat{\sigma_{\mathcal{P}}})$

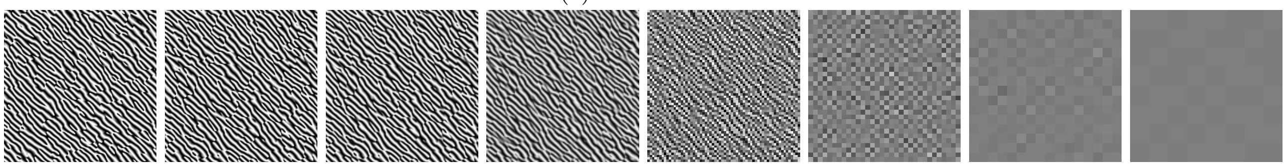


(c) Error $||\widehat{S} - \overline{S}||$

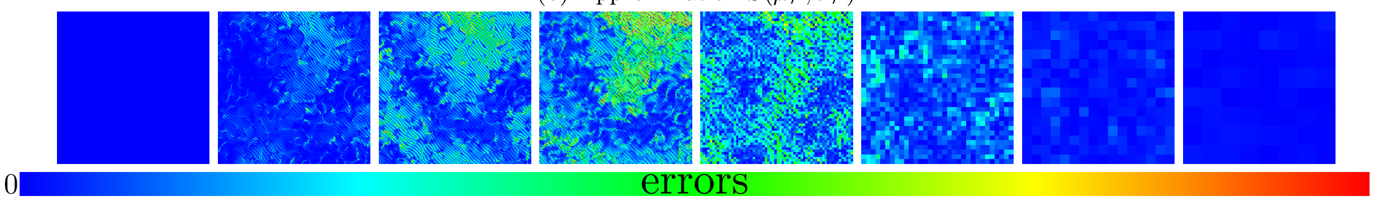
Figure 16: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with exact computation of the mean and approximation of the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.



(a) Reference $\overline{S}$



(b) Approximation $\widehat{S}(\widehat{\mu_{\mathcal{P}}}, \widehat{\sigma_{\mathcal{P}}})$



(c) Error $||\widehat{S} - \overline{S}||$

Figure 17: Approximation $\widehat{S}$ of the filtering $\overline{S}$ with approximation of the mean and the standard deviation. From left to right: footprint size equal to 1 (input), 2, 4, 8, 16, 32, 64, 128. Size of the input: 1024×1024.

References

[Bur19] Brent Burley. On histogram-preserving blending for randomized texture tiling. *Journal of Computer Graphics Techniques*, 8(4), 2019.

[DH19] Thomas Deliot and Eric Heitz. Procedural stochastic textures by tiling and blending. *GPU Zen*, 2, 2019.

[HN18] Eric Heitz and Fabrice Neyret. High-performance by-example noise using a histogram-preserving blending operator. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(2):1–25, 2018.