

# Compte rendu : Bruits à variation spatiale en temps réel

Arthur Chateauneuf

April 16, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Contexte du stage . . . . .	2
1.2	Présentation du sujet et des problématiques . . . . .	2
1.2.1	Processus Stochastique . . . . .	2
1.2.2	Variation Spatiale . . . . .	2
1.2.3	Temps réel: utilisations et implications . . . . .	2
1.2.4	Parrallélisation . . . . .	2
1.2.5	Complexité mémoirelle et calculatoire sur GPU . . . . .	2
1.2.6	Intégration analytique . . . . .	2
<b>2</b>	<b>Travaux précédents</b>	<b>2</b>
2.1	State of the Art in Procedural Noise Functions . . . . .	2
2.2	Random Phase Textures: Theory and Synthesis . . . . .	2
2.3	Cyclostationary Gaussian noise: theory and synthesis . . . . .	2
<b>3</b>	<b>Spike Noise: Approche cas par cas</b>	<b>2</b>
3.1	Concept . . . . .	2
3.2	Non Stationarité . . . . .	2
3.3	Intégration . . . . .	2
<b>4</b>	<b>Mappage procédural de priorité : Approche générale de mélange</b>	<b>3</b>
4.1	Mélange de texture : concept et introduction . . . . .	3
4.2	L'opérateur Mix Max . . . . .	3
4.3	Apports pour les processus stochastiques en niveau de gris . . . . .	3
4.4	Estimation de la priorité d'un processus stochastique . . . . .	4
4.5	Nouvel opérateur d'influence équilibré . . . . .	5
4.6	Démonstration de l'intégration . . . . .	6
4.7	Conclusion . . . . .	6

## 1 Introduction

- 1.1 Contexte du stage
- 1.2 Présentation du sujet et des problématiques
  - 1.2.1 Processus Stochastique
  - 1.2.2 Variation Spatiale
  - 1.2.3 Temps réel: utilisations et implications
  - 1.2.4 Parrallélisation
  - 1.2.5 Complexité mémoirelle et calculatoire sur GPU
  - 1.2.6 Intégration analytique

## 2 Travaux précédents

- 2.1 State of the Art in Procedural Noise Functions
- 2.2 Random Phase Textures: Theory and Synthesis
- 2.3 Cyclostationary Gaussian noise: theory and synthesis

## 3 Spike Noise: Approche cas par cas

- 3.1 Concept
- 3.2 Non Stationarité
- 3.3 Intégration

## 4 Mappage procédural de priorité : Approche générale de mélange

### 4.1 Mélange de texture : concept et introduction

La création d'environnements virtuels peut nécessiter des transitions d'aspects sur une même surface. Cependant, les besoins et les restrictions ne permettent pas tout le temps de pouvoir réaliser cet effet manuellement. Dans ce cadre, un mélange est utilisé afin d'utiliser plusieurs textures sur la surface.

Un mélange entre plusieurs textures est composé d'une fonction d'influence (ou transparence) pour chaque entrée, notée  $\alpha_x$  pour l'entrée  $x$ , ainsi que d'une fonction de mélange, notée  $M$ . Un tel processus crée une donnée de sortie résultant d'un mélange biaisé entre toute ces entrées. Le biais du mélange est dicté par la fonction d'influence. Ce dernier doit être borné, de plus, la somme de l'influence de toutes les entrées doit être constante.

$$\alpha_{x_0} \in [0, 1] \quad (1)$$

$$\sum_i \alpha_{x_i} = 1 \quad (2)$$

Lorsque l'influence d'une entrée tend vers le maximum, la fonction de mélange doit alors tendre vers cette entrée.

$$\lim_{\alpha_{x_0} \rightarrow 1} M(x_0, \alpha_{x_0}, \dots, x_n, \alpha_{x_n}) = x_0 \quad (3)$$

La forme la plus répandue de cette opération est le mélange linéaire s'exprimant sous la forme :

$$M^L = \alpha_{x_0} x_0 + \dots + \alpha_{x_n} x_n \quad (4)$$

Si l'on considère les entrées comme des variables aléatoires  $V_i$  possédant une espérance  $\mathbb{E}[V_i]$  nous pouvons alors utiliser un mélange préservant la variance [FS18], s'exprimant :

$$M^{cov} = M^L(\mathbb{E}[V_0], \alpha_{V_0}, \dots) + \frac{\sum \alpha_{V_i} (V_i - \mathbb{E}[V_i])}{\sqrt{\sum \alpha_{V_i}^2}} \quad (5)$$

Si les entrées possèdent une distribution gaussienne, l'histogramme de ces dernières est alors également préservé. Maintenir les propriétés statistiques des entrées permet une meilleure qualité visuelle du mélange, en évitant les biais visibles sur une transition [FS18].

Dans le cas d'un mélange entre deux entrées, nous ne notons qu'un seul biais  $A = \alpha_{x_1}$  car le second peut être déduit de façon triviale par  $\alpha_{x_0} = 1 - \alpha_{x_1}$ . La fonction de mélange respecte alors les propriétés :

$$\lim_{A \rightarrow 0} M(x_0, x_1, A) = x_0 \quad (6)$$

$$\lim_{A \rightarrow 1} M(x_0, x_1, A) = x_1 \quad (7)$$

Tous les états possibles du mélange peuvent ainsi être représentés par une seule fonction d'influence.

### 4.2 L'opérateur Mix Max

Romain Fournier et Basile Sauvage ont présentés en 2024 l'opérateur d'influence de mélange MixMax [HN24]. Ce dernier répond à un besoin d'obtenir en temps réel des transitions de textures à la fois nettes et sensibles à leur contenu. Cette méthode se base sur l'existence d'une carte de priorité, donnée pour chaque entrées. Ce mélange est analytiquement filtrable, paramétrable et s'exprime sous la forme :

$$A = \Phi \frac{(S_0(\mathbb{P}) + P_0(\mathbb{P})) - (S_1(\mathbb{P}) + P_1(\mathbb{P}))}{\sqrt{\sigma_0^2 + \sigma_1^2 + \lambda_0^2 + \lambda_1^2}} \quad (8)$$

Avec  $S_i(\mathbb{P})$  la moyenne de l'entrée  $i$  sur la zone  $\mathbb{P}$ ,  $P_i(\mathbb{P})$  la moyenne de la carte de priorité de l'entrée  $i$  sur la même zone,  $\sigma_i$  l'écart-type de l'entrée  $i$ ,  $\lambda_i$  le décalage de la variance de l'entrée  $i$  (utilisée afin de contrôler la netteté de la transition) et  $\Phi$  un facteur de lissage calculé à partir de la différence entre les priorités (en application, il est approximé à partir d'une loi gaussienne).

### 4.3 Apports pour les processus stochastiques en niveau de gris

Une transition entre des processus stochastiques procéduraux permet de créer une infinité de bruits variant spatialement. Cependant, les mélanges existants pour ces processus, comme l'influence linéaire, apportent une pauvre qualité visuelle. L'application d'un mélange sensible au contenu dans ce cadre permettrait d'obtenir des résultats temps réels similaires à ceux de l'opérateur Mix-Max tout en exploitant les propriétés de tels processus, notamment en étant non borné. Un tel apport, à notre connaissance, n'a encore jamais été étudié.

Dans le papier de Fournier et Sauvage, cette fonction d'influence n'est utilisée que sur des textures matricielles possédant une carte de priorité déjà existante. Nous proposons une fonction de calcul temps réel de la priorité d'un processus stochastique, ainsi qu'un nouvel opérateur d'influence inspiré des principes du Mix-Max plus adapté à notre cas d'utilisation.

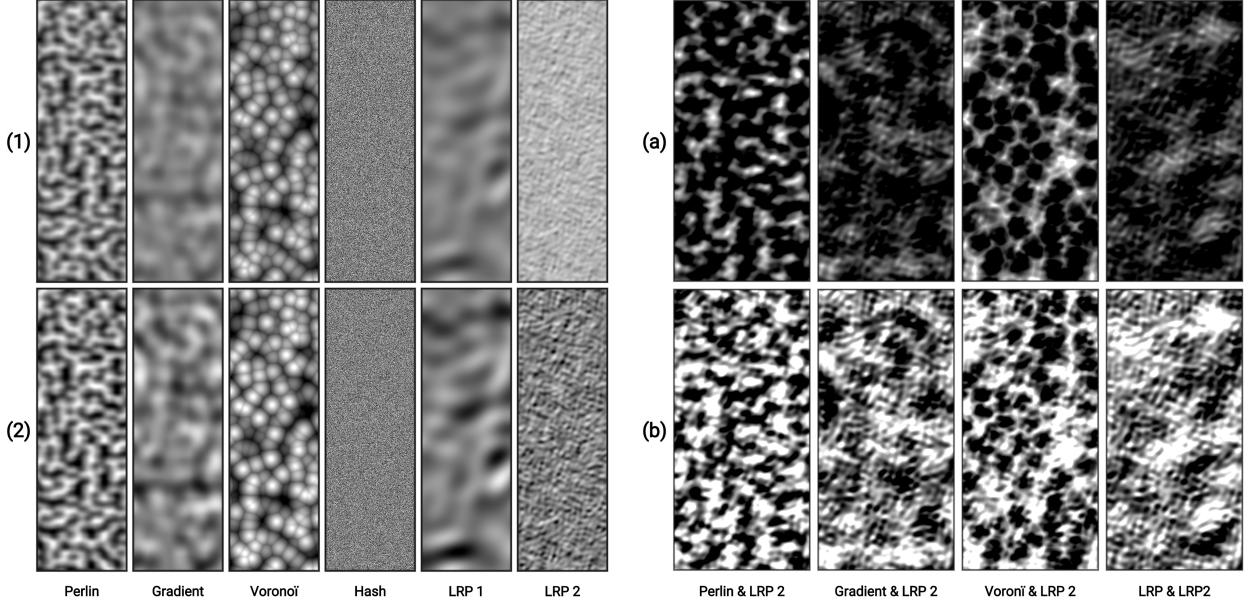


Figure 1: Comparaison entre les processus stochastiques originaux (1) et les résultats donnés par la fonction de priorité (2). Les niveaux de gris originaux (1) sont compris entre 0 (noir) et 1 (blanc) tandis qu'ils sont représentés la ligne (2) dans l'intervalle -1 (noir) à +1 (blanc).

Figure 2: Résultats de notre fonction d'influence avec un biais d'entrée de 0.5 sans correction de la priorité (a) et après réhaussement de la priorité (b). Nous observons à la ligne (b) une fréquence de présence similaires entre les processus, peut importe leur propriétés statistiques.

#### 4.4 Estimation de la priorité d'un processus stochastique

Un processus stochastique possède une intensité à chaque endroit de l'espace, une carte de priorité sensible à ce dernier possèderait une intensité directement corrélée à l'intensité du bruit. Une première approche pourrait être d'utiliser l'intensité du bruit en tant que priorité. Les motifs et composantes du bruit ressortiraient ainsi lors du mélange. Le problème de cette approche est qu'un bruit très lumineux ou très sombre auraient une priorité disproportionnée dans le mélange, comme visible dans la figure 2. Des valeurs extrêmes peuvent même entraîner la carte de priorité à briser les priorités vus à l'équation (3).

Nous proposons une solution, notée  $\mathcal{P}$ , de priorité procédurale utilisant les propriétés statistiques d'un processus stochastique donné afin d'en extraire une carte de priorité respectant l'intensité du bruit, le tout en ayant une intensité correctement répartie entre  $[-1, 1]$  pour n'importe quelle entrée. Cette méthode s'apparente à l'approximation d'un étalement d'histogramme utilisant l'espérance  $\mathbb{E}$  et l'écart-type  $\sigma$  d'une variable aléatoire  $V$ .

$$\mathcal{P}^{\pm 1}(V) = \pm \begin{cases} -1, & \text{Si } V < \mathbb{E}[V] - \sqrt{\sigma[V]} \\ +1, & \text{Si } V > \mathbb{E}[V] + \sqrt{\sigma[V]} \\ \frac{V - \mathbb{E}[V]}{\sqrt{\sigma[V]}}, & \text{Sinon} \end{cases} \quad (9)$$

Cette méthode donnera des bornes d'étalements, ie les bornes de l'intervalle d'intensité dans l'entrée qui sera converti à  $[-1, 1]$  dans la sortie, centrés sur  $\mathbb{E}[V]$  et d'intervalle  $2\sqrt{\sigma[V]}$ .

$$\lim_{V \rightarrow \mathbb{E}[V] - \sqrt{\sigma[V]}} \mathcal{P}(V) = -1 \quad (10)$$

$$\lim_{V \rightarrow \mathbb{E}[V] + \sqrt{\sigma[V]}} \mathcal{P}(V) = +1$$

Dans la figure 1, nous observons que la correction de contraste obtenue avec cette méthode est équilibré sur tous les processus que nous avons pu tester.

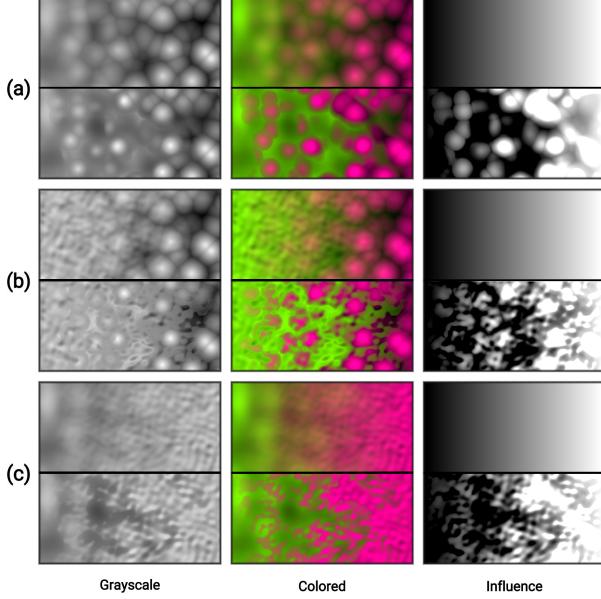


Figure 3: Résultats du mélange entre Voronoï et Gradient (a), Voronoï et LRP2 (b) ainsi que LRP2 et Gradient (c). La moitié supérieure des cellules montrent un gradient d'influence linéaire  $\alpha$  et la partie inférieure un gradient utilisant la fonction  $A_{0.5}^c$  en influence.

#### 4.5 Nouvel opérateur d'influence équilibré

Nous proposons un opérateur d'influence  $\mathcal{A}$  entre deux processus stochastiques  $V_0$  et  $V_1$  basé sur les principes du MixMax, simplifiant la formule donnée dans le papier original [HN24] grâce aux propriétés obtenues par notre fonction de priorité procédurale.

$$\begin{aligned} \mathcal{L}_\alpha(x) &= \begin{cases} 0, & \text{Si } x < a \\ 1, & \text{Si } x > b \\ \frac{x}{a} - \frac{1}{2} & \text{Sinon} \end{cases} \\ \rho &= \frac{\mathcal{P}_{V_0}^{\pm 1} + \mathcal{P}_{V_1}^{\pm 1}}{2} \\ |\rho| &= \frac{|\mathcal{P}_{V_0}| + |\mathcal{P}_{V_1}|}{2} \\ \mathcal{A}_s &= \mathcal{L}^s(\rho + (2\alpha - 1)(s + |\rho|)) \end{aligned} \quad (11)$$

Cette fonction utilise un biais linéaire  $\alpha$  ainsi qu'une

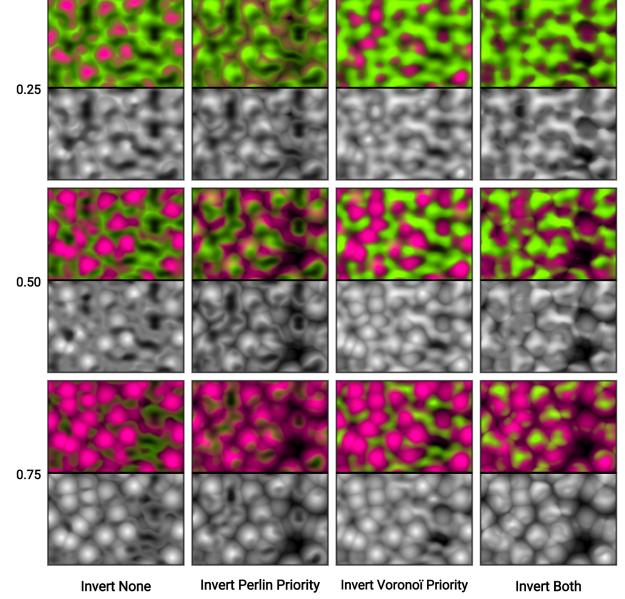


Figure 4: Résultats du mélange utilisant  $A_{0.5}^c$  entre Voronoï et Perlin en fonction du biais d'entrée  $\alpha$  (lignes) et des directives d'inversions de priorités (colonnes). Nous remarquons que lorsqu'une seule des priorités est inversée, des biais statistiques deviennent visibles.

valeur de lissage  $s$  strictement positive. Nous montrons l'impact de ces paramètres sur le mélange dans la figure 6.

Chacune des deux fonctions de priorité peut être inversée. Les différences visuelles qu'impliquent ces inversions sont visibles sur la figure 4, où l'équilibre du contraste dans les zones de transition varie en fonction du paramétrage.

Lorsque nous comparons la moyenne des présences de chaque bruit en fonction de  $\alpha$ , nous remarquons des imprécisions. L'influence de sortie suit une transition douce lorsque l'écart des priorités à la moyenne est grand. Le principe de transition douce est un outil régulièrement utilisé dans la création de mouvements ou de transitions agréables à l'œil humain. Cependant, ce phénomène entraîne une disparité de présence des bruits dans notre mélange. Nous proposons une version corrigée de notre méthode prenant en compte ce biais et le corrigeant à l'aide de la fonction de transition douce inverse  $\mathcal{S}^{-1}$  :

$$\mathcal{A}_s^c = |\rho| \mathcal{S}^{-1}(\mathcal{A}_s) + (1 - |\rho|) \mathcal{A}_s \quad (12)$$

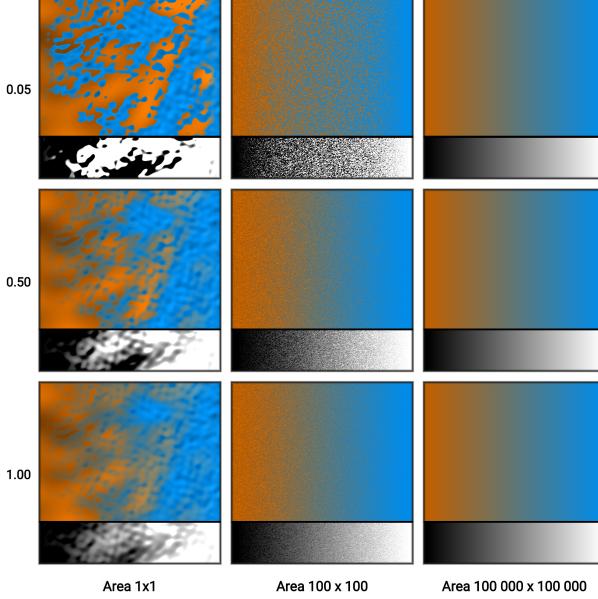


Figure 5: Résultats de transition utilisant  $\mathcal{A}_s^c$  entre deux processus analytiquement filtrable LRP1 et LRP2 en fonction du lissage  $s$  (lignes) et de la taille de la zone  $\mathbb{P}$ . Les entrées du mélange sont colorées pour une meilleure visibilité. La valeur de l'influence est également visible dans le quart inférieur de chaque cellule.

#### 4.6 Démonstration de l'intégration

Comme expliqué dans la section 1.2.6, le rendu d'une surface en temps réel nécessite un filtrage analytique de sa texture. Dans notre cas, nous notons  $V(\mathbb{P})$  l'empreinte d'une variable aléatoire  $V$  sur la zone  $\mathbb{P}$  de taille  $||\mathbb{P}||$ . Selon l'équation 11, nous obtenons :

$$\begin{aligned} \lim_{||\mathbb{P}|| \rightarrow +\infty} \mathcal{P}_V(\mathbb{P}) &= 0 \\ \lim_{||\mathbb{P}|| \rightarrow +\infty} \mathcal{A}_s^c(\mathbb{P}) &= \alpha(\mathbb{P}) \\ \lim_{||\mathbb{P}|| \rightarrow +\infty} \mathcal{A}_s(\mathbb{P}) &= \alpha(\mathbb{P}) \end{aligned} \quad (13)$$

Sur les très larges zones, notre mélange est ainsi correctement filtré analytiquement. Cependant, comme vu précédemment, sur des petites zones, l'espérance de la présence des entrées n'est pas linéaire avec  $A_s$ . La correction  $A_s^c$  permet ainsi d'obtenir un filtrage très précis.

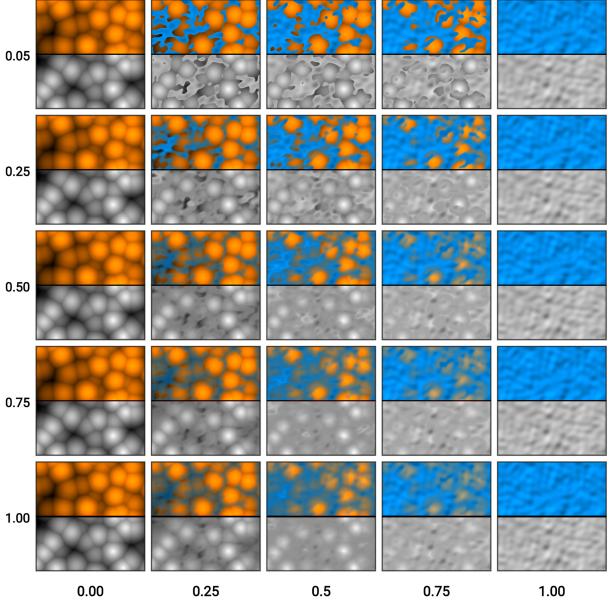


Figure 6: Résultats du mélange entre Voronoï et LRP2 utilisant  $\mathcal{A}_s^c$  en fonction de  $s$  (lignes) et du biais linéaire d'entrée  $\alpha$  (colonnes). Les priorités des deux entrées sont inversées.

$$\begin{aligned} \int_{\omega} \lim_{||\mathbb{P}|| \rightarrow 0} \mathcal{A}_s(\mathbb{P}) &\approx \mathcal{S}(\alpha(\mathbb{P})) \\ \int_{\omega} \lim_{||\mathbb{P}|| \rightarrow 0} \mathcal{A}_s^c(\mathbb{P}) &\approx \alpha(\mathbb{P}) \end{aligned} \quad (14)$$

Lorsque la valeur de lissage  $s$  est très petite ( $s < 0.05$ ), l'écart de cette approximation avec le filtrage réel s'accroît. Cependant, nous n'avons pas trouvé de marge d'erreure significative dans notre implémentation pour toutes les valeurs de  $s$  supérieures à 0.05.

#### 4.7 Conclusion

La méthode que nous proposons permet de créer des transitions nettes et sensibles au contenu entre n'importe quel processus stochastique. La formule utilisé est analytiquement filtrable lorsque toutes les entrées le sont également. Cependant, certaines limitations persistent, comme les imprécisions de filtrage pour des transitions très nettes.

Un tel outil ouvre un large champ de possibilités pour la création de processus stochastiques à variation spatiale, le tout en temps réel et avec un très faible impact en calcul et en mémoire.

## References

- [FS18] Romain Fournier and Basile Sauvage. “Mix-Max: A Content-Aware Operator for Real-Time Texture Transitions”. In: (2018). URL: <https://inria.hal.science/hal-01824773v1/document>.
- [HN24] Eric Heitz and Fabrice Neyret. “High-Performance By-Example Noise using a Histogram-Preserving Blending Operator”. In: *COMPUTER GRAPHICS forum* (2024). URL: <https://onlinelibrary.wiley.com/doi/10.1111/cgf.15193>.