

## TER : Synthèse et rendu fractal de textures

### Contexte

Ce TER est plutôt orienté *recherche appliquée*.

Le pavage périodique d'une texture sur une surface consiste à créer un échantillon de textures qui soit périodique (voir : *seamless textures*), puis à répéter cet échantillon sur la surface. Le problème du pavage périodique est que la surface présente des artefacts de répétitions, qui cassent l'immersion dans le monde virtuel contenant de telles surfaces.

La synthèse de textures temps réel et par l'exemple permet d'étendre une texture sur une surface à partir d'un échantillon périodique de texture (c'est pour ça qu'on dit qu'elle est par l'exemple) en introduisant de l'aléa en temps réel (c'est pour ça qu'elle est dite temps réel), qui permet de résoudre ce problème.

### Problème

Le problème que nous cherchons à résoudre est que les algorithmes de synthèse temps réel actuellement proposés permettent de synthétiser efficacement des surfaces texturées de taille non bornées [HN18, DH18, LSD23], mais ne permettent pas de synthétiser des surfaces dont l'apparence est continue (c'est-à-dire qu'on peut théoriquement zoomer à l'infini sur une surface sans voir apparaître de pixels). C'est un problème important, car il empêche les surfaces synthétisées d'avoir un aspect naturel lorsqu'elles sont scrutées de près. Il est cependant difficile à résoudre étant donné que les textures utilisées comme exemple sont des images matricielles, donc par définition discrètes.

### Travail demandé

Vous étudierez la possibilité de texturer une surface fractalement (approche proposée dans le shadertoy de Xor <https://www.shadertoy.com/view/mds3R4>) en la reprogrammant. Il faut voir si et dans quels cas cette approche permet de résoudre le problème : pour quelles textures, pour quel genre d'apparence, dans quelles conditions d'éclairage et de vue, s'il faut modifier la technique et comment, ou si ça ne fonctionnera pas, justifier pourquoi. Ensuite, vous combinerez le rendu fractal de textures avec la synthèse de Heitz et Neyret [HN18] et étudierez le résultat. Pour que le travail ne se complexifie pas trop, le transfert d'histogramme proposé par Heitz et Neyret [HN18] ne devra pas être implémenté (seule la préservation de variance qu'ils proposent est généralement utilisée dans l'industrie de toute façon).

### Références

- [DH18] Thomas Deliot and Eric Heitz. Procedural stochastic textures by tiling and blending. *GPU Zen 2 : Advanced Rendering Techniques*, 2018.
- [HN18] Eric Heitz and Fabrice Neyret. High-performance by-example noise using a histogram-preserving blending operator. *Eurographics Symposium on High-Performance Graphics 2018*, 2018.
- [LSD23] Nicolas Lutz, Basile Sauvage, and Jean-Michel Dischler. Preserving the Autocovariance of Texture Tilings Using Importance Sampling. *Computer Graphics Forum*, 2023.