Université de Montpellier
Faculté des sciences

Date : Mardi 2 décembre 2025
Enseignant : Nicolas Lutz

TER : Synthèse de texture volumique à partir d'un exemple surfacique

Contexte

Ce TER est plutôt orienté *recherche appliquée*.

En informatique graphique, l'augmentation de la taille des mondes virtuels rend difficile la création et le stockage de la géométrie et de l'apparence des objets peuplant ces mondes. Pour adresser ce problème, la génération procédurale forme une famille d'algorithmes qui permettent de créer automatiquement les contenus d'un monde virtuel. La synthèse de textures temps réel fait partie de cette famille et permet de générer l'apparence de surfaces de taille non bornée voire infinies sans augmenter l'occupation mémoire.

La synthèse de textures temps réel par l'exemple, en particulier, consiste à étendre un petit échantillon de texture sur la surface à l'aide d'un algorithme contenant de l'aléa, sans les répétitions alignées que causeraient le fait de répéter cet échantillon à l'infini sur la surface (une technique couramment utilisée dans le jeu vidéo qu'on appelle "pavage périodique").

En 2025, Brian Delvigne a effectué un stage au LIRMM et a proposé un prototype d'algorithme permettant de synthétiser un volume à partir d'une texture surfacique, en adaptant la synthèse de textures de Heitz et Neyret [HN18] pour faire en sorte que chaque face soit exprimée par un empilement de synthèses, ce qui permet de décrire un volume. Ce prototype permet de déplacer une surface dans l'espace pour explorer un volume infini, comme une loupe.

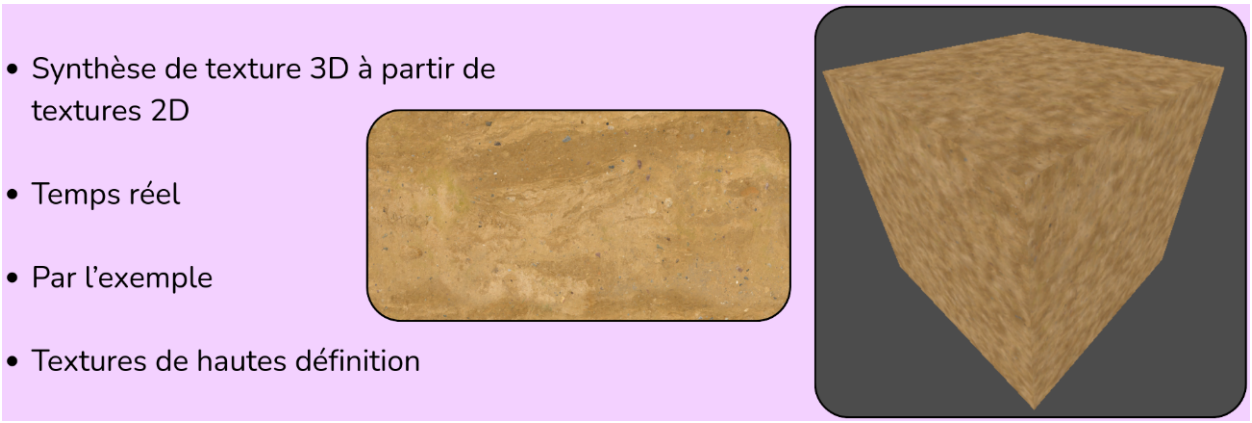

FIGURE 1 – Capture d'une slide de la présentation de Brian Delvigne. Une texture volumique (droite) est synthétisée à partir d'une texture surfacique (gauche). Cette représentation permettrait de couper le cube en deux et d'avoir une définition de l'apparence qui soit cohérente sur la tranche.

Problème

Le problème du travail de Brian Delvigne est qu'il se limite à la projection du volume sur une surface rastérisée, puisque le shader a lui-même été programmé comme décrivant l'apparence d'une surface rastérisée. Cette limite rend impossible le fait de représenter des surfaces semi-transparentes, qui sont habituellement rendues grâce au *ray marching*, une technique de lancer de rayons qui consiste à explorer un volume en calculant à chaque pas les contributions de ce volume.

Travail demandé

Vous commencerez par implémenter un *ray marching*, en temps réel ou non, peu importe le support de programmation. Pour montrer que votre moteur fonctionne, vous représenterez des formes simples basées sur

des champs de distance signée avec votre *ray marching* (voir les travaux d'Inigo Quilez [IQ]). Ensuite, vous réimplémenterez la synthèse de Brian Delvigne, en l'adaptant à des exemples de textures semi-transparentes.

Références

[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.

[IQ] `https://iquilezles.org`.