

RAPPORT TP3 INF8702

Adrien Logut (1815142), Chathura Namalgamuwa (1815118)

Question 1

On vous a fourni la fonction `lineariserProfondeur()`. Expliquez ce qu'accomplit cette fonction.

La matrice de projection transforme la coordonnée de profondeur z en z' et la rend proportionnelle à $\frac{1}{z}$ (le facteur de proportionnalité dépend de z_{near} et z_{far} , les valeurs extrêmes de notre matrice de projection).

Cependant pour calculer notre distance, on veut que la différence entre `depth` et `center_depth` soit linéaire. Il faut donc faire repasser notre composante de profondeur en composante linéaire. La fonction `lineariserProfondeur()` s'en charge, en utilisant z_{near} et z_{far} .

Question 2

On vous a aussi fourni la fonction `FiltreGaussien()`. Expliquez les détails de son fonctionnement.

La fonction `filtreGaussien` effectue une convolution entre la matrice gaussienne 5x5 de déviation standard $\sigma = 2.5$ et la matrice de pixels représentant la texture au point (x, y) .

Question 3

Cette implémentation naïve du champ de profondeur effectue combien d'échantillonnage (sampling) de textures par fragment du quad plein écran ? Donner les détails sur une autre technique utilisée réduisant le nombre d'échantillonnage pour une même portée.

On a une matrice gaussienne 5x5, ce qui veut dire que pour un fragment, on a besoin des 25 pixels qui sont autour de lui. On a donc 25 échantillonnages à faire pour chaque fragment, à chaque appel à `FiltreGaussien()`. Sachant qu'on l'appelle deux fois (avec deux étendues différentes), on est à 50 échantillonnages par fragment.

De part la propriété de la gaussienne, on peut séparer le filtre bi-dimensionnel en 2 filtres uni-dimensionnels. Ainsi, on peut filtrer tout d'abord sur x (5 échantillonnages/fragment) puis filtrer le résultat sur y (5 échantillonnages/fragment). On passe donc de 25 échantillonnages/fragment à 10 échantillonnages/fragment. Ce qui est un gain conséquent.

Question 4

Avec notre implémentation présente du IBL, le gazon n'est pas visible dans les réflexions de l'environnement. Comment pourriez-vous régler ce problème en utilisant des FBOs ? Considérez ici un seul modèle 3D présent sur le gazon.

Le FBO permet de "dessiner dans un buffer" plutôt que de dessiner dans la fenêtre. C'est utile si l'on veut faire un miroir ou encore une caméra de surveillance qui filme une autre scène. Pour que le gazon soit visible dans les réflexions de l'environnement, on peut rendre le gazon dans un FBO pour qu'il devienne une texture (comme on a fait pour le flou). Ici notre gazon est plat, on peut donc à partir de là, construire une "skybox" qui sera un cube avec sur une des faces notre gazon rendu et ailleurs "rien" (aucune composante à rajouter).

A partir de là, on peut réappliquer la même technique que pour notre skybox et rajouter le gazon en tant que réflexion de l'environnement.