

SI3D / GPU

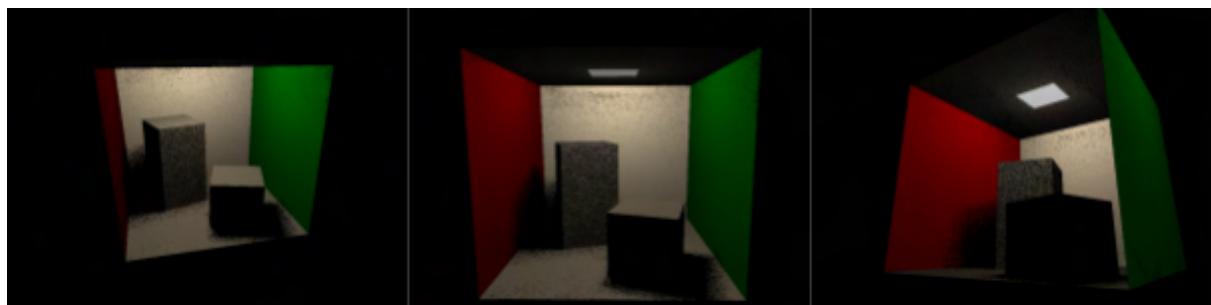
[Lien GitHub](#)

Ray-tracing

Nous avons implémenté une application interactive. L'utilisateur est muni d'un spray lui permettant d'envoyer des rayons sur la scène. Le point d'intersection est calculé puis stocké pour être ensuite dessiné selon le point de vue.

Les informations nécessaires à l'illumination d'un point sont précalculées puis l'illumination finale est calculée dans le fragment shader en fonction de la vue.

Cela est seulement possible pour les termes n'incluant pas une source lumineuse et le point de vue à la fois. Cependant la scène ne dispose pas de tels termes (ex: réflexion).



3 points de vues différents du même nuage de points (8 millions de points) de la Cornell Box.

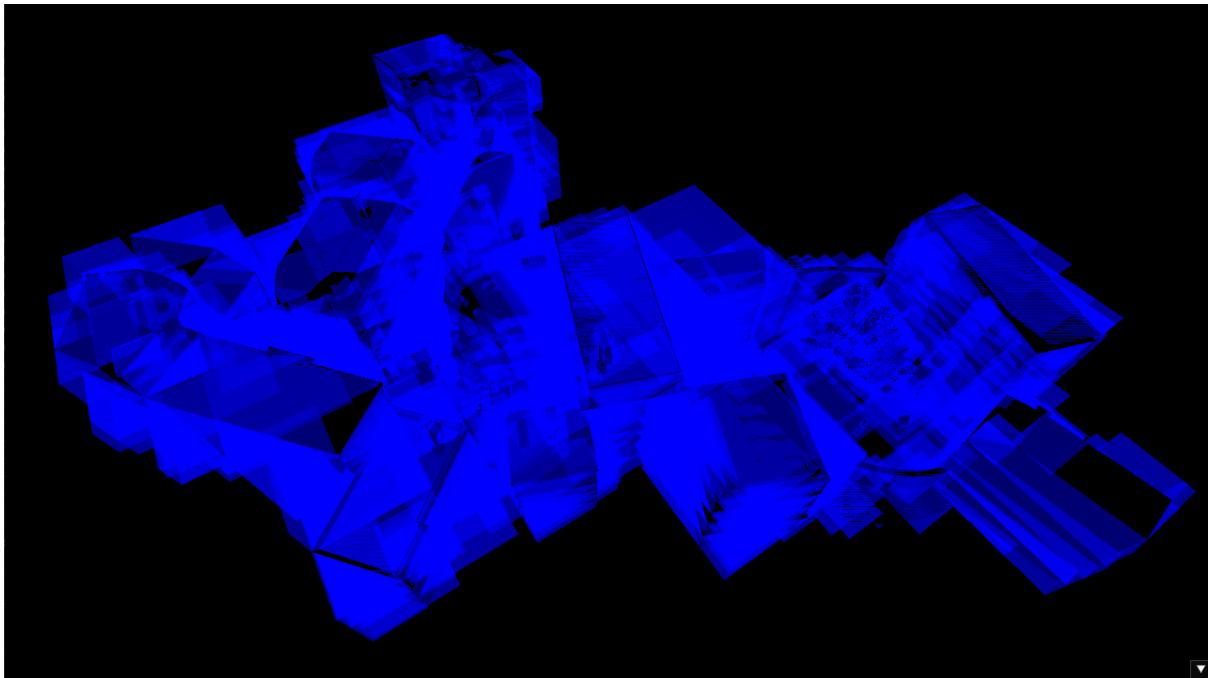
L'illumination est calculée en fonction d'un terme lambertien du point par rapport à la source lumineuse et du point par rapport à l'observateur. Un terme d'atténuation est aussi calculé pour la source et l'observateur.

Les sources lumineuses sont tirées aléatoirement (avec la formule 18 du Global Illumination Compendium). L'émissivité et l'albédo sont récupérés dans le matériel du triangle.

Rasterisation

Optimisation

Découpage de la scène



Découpage de la scène en 8x8x8.

La scène est découpée par une grille régulière en 8x8x8 sur les centroïdes des triangles. Les 1077 boîtes englobantes obtenues sont ensuite agrandies pour contenir les sommets des triangles.

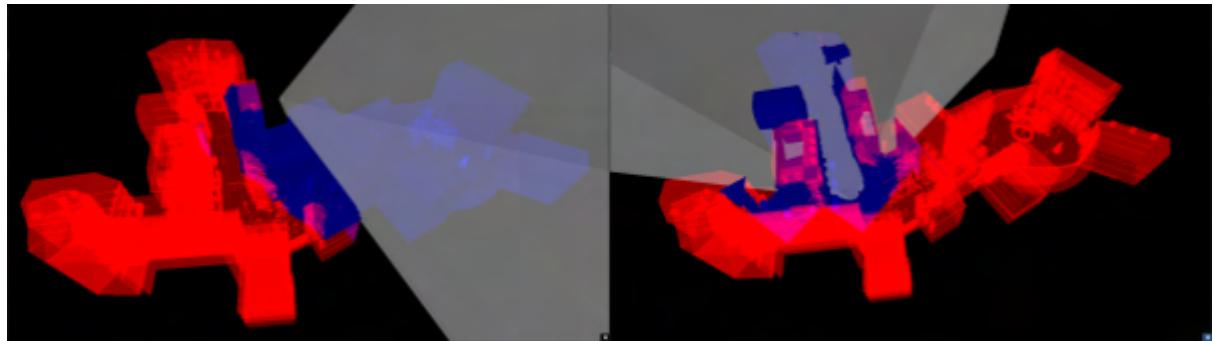
Un buffer de commandes est créé pour chacun des groupes de triangles contenues dans les boîtes englobantes. Ce buffer est utilisé avec `glMultiDrawElementsIndirect` pour dessiner toute la scène en une seule commande OpenGL.

Frustum culling

Le frustum culling a été implémenté sur GPU.

La boîte englobante de chaque objet est testée, selon le théorème des axes séparés, avec la boîte englobante du frustum dans le repère monde et dans le repère projectif. Si les deux boîtes sont séparées par un plan alors l'objet est filtré.

Le compute shader prend donc en entrée les commandes à filtrer, les boîtes englobantes dans le repère monde des objets, une transformation du repère monde au repère projectif, la boîte englobante du frustum dans le repère monde et dans le repère projectif; et retourne en sortie les commandes qui passent le test.



En blanc, le cone de vue; en rouge, les objets culls; en bleu, les objets conservés.

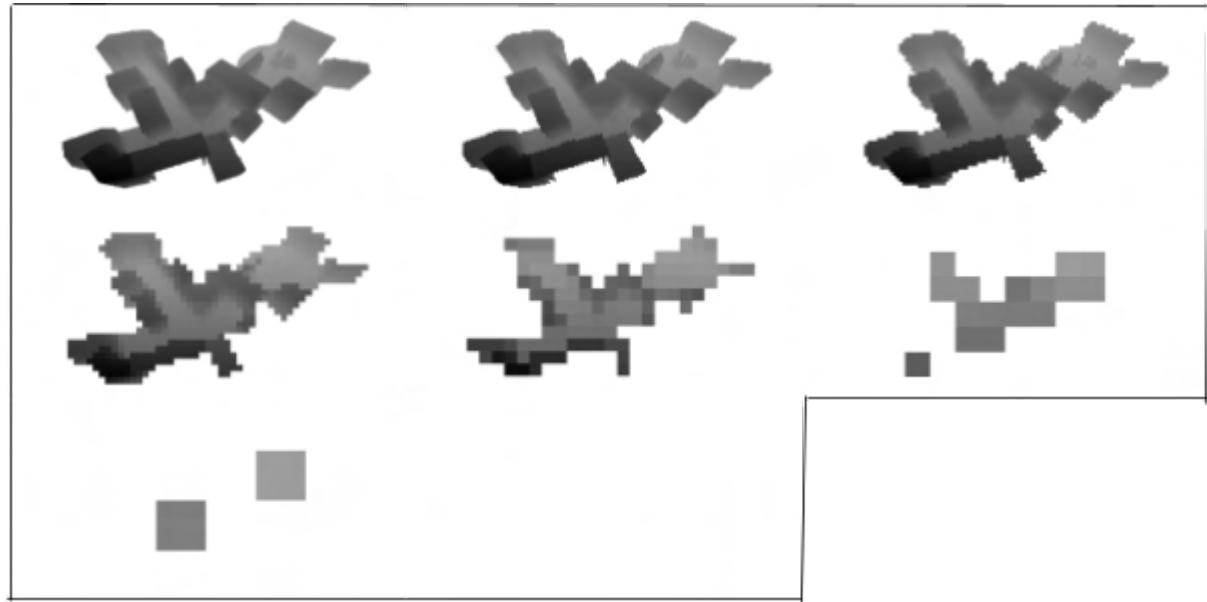
Sur l'image de gauche, 101 objets sur 208 (49.6%) passent le test.

Sur l'image de droite, 59 objets sur 208 (28.3%) passent le test.

Occlusion culling

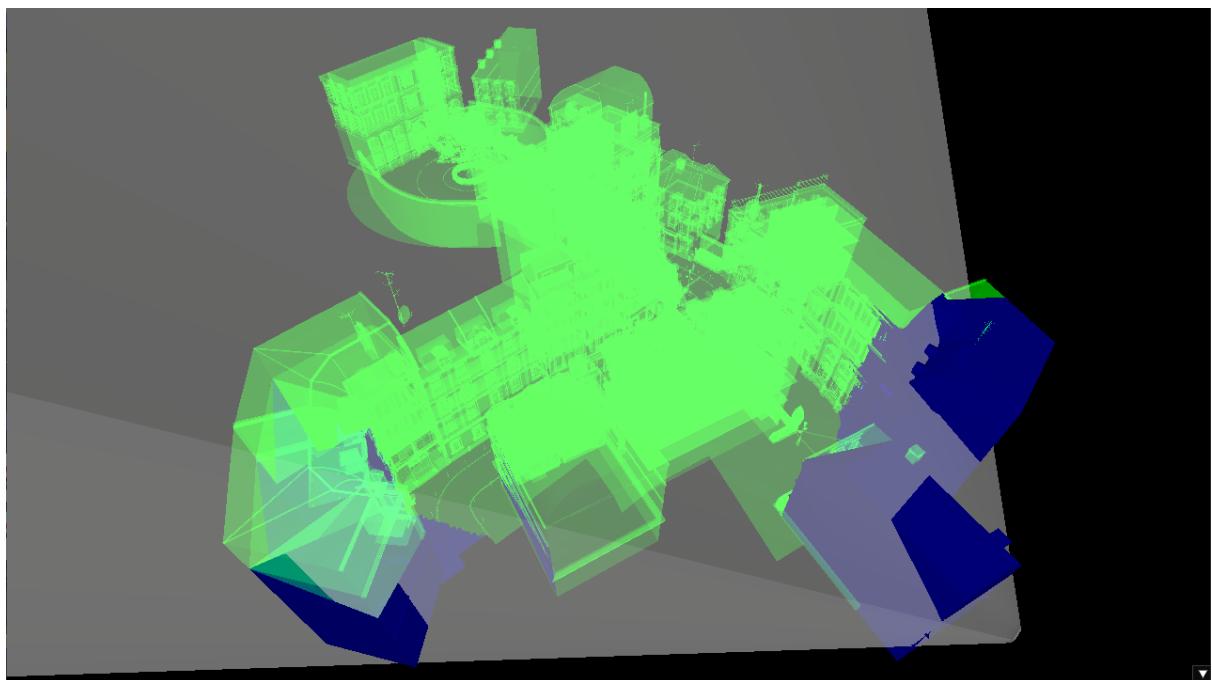
L'occlusion culling avec Z-buffer hiérarchique a été implémenté sur GPU.

Une première passe de profondeur est effectuée sur les objets passant le frustum culling. La résolution du framebuffer utilisée est de 521x256 et ne comporte qu'un depth buffer.



8 niveaux du Z-buffer hiérarchique.

Le depth buffer est ensuite réduit successivement sur GPU. À chaque niveau, un quad de pixels est remplacé par la valeur maximale. La résolution est donc divisée par 2 à chaque niveau. La résolution du huitième niveau est de 4x2.



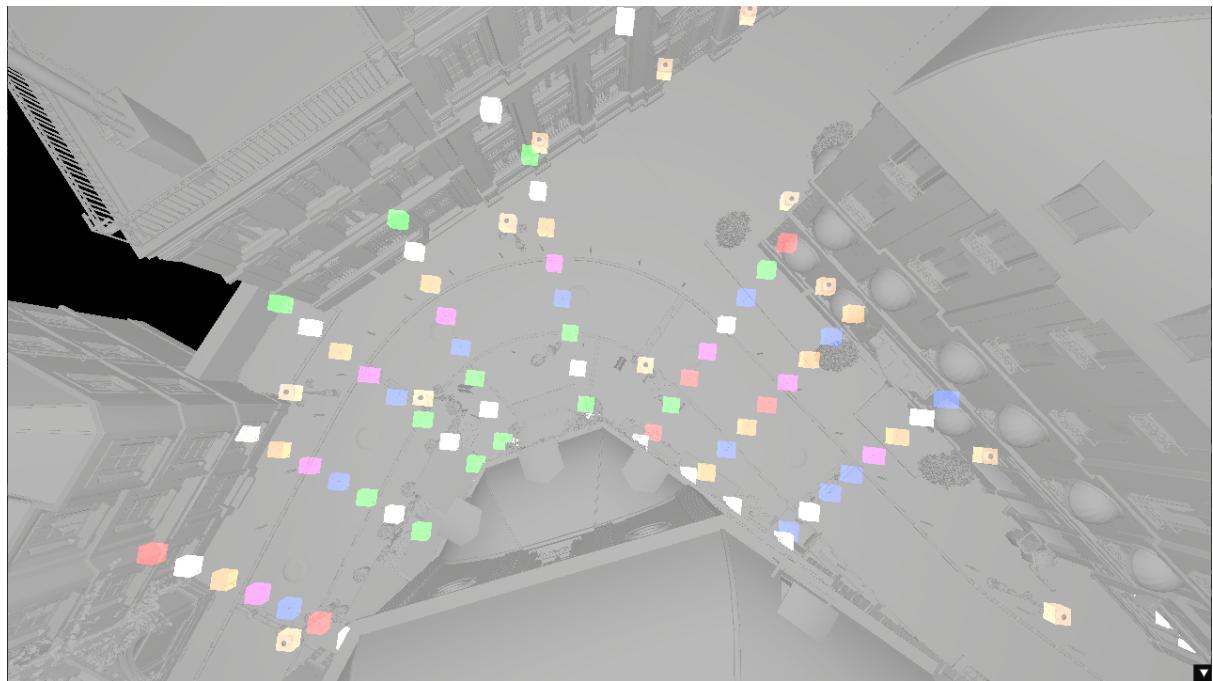
En blanc, le cone de vue; en vert, les objets culles; en bleu, les objets conservés.

Sur l'image de gauche, 44 objets sur 208 (21.1%) passent le test.

Light culling

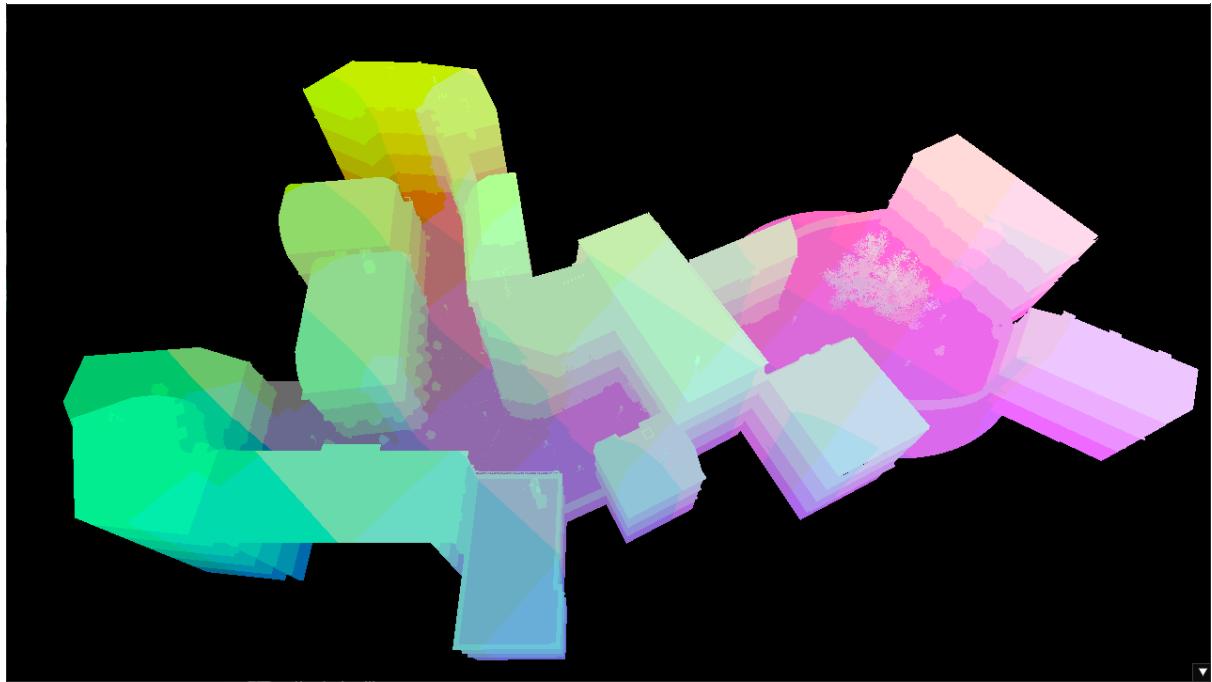
La liste des lumières est récupérée depuis le fichier OBJ en assemblant les triangles émissifs de même matériau et de proximité suffisante. On obtient 170 sources lumineuses.

Cependant le fichier décrit certaines sources en plusieurs fois. En agglomérant les différents morceaux de ces sources, on descend à 130 sources lumineuses.



Emplacement des sources lumineuses.

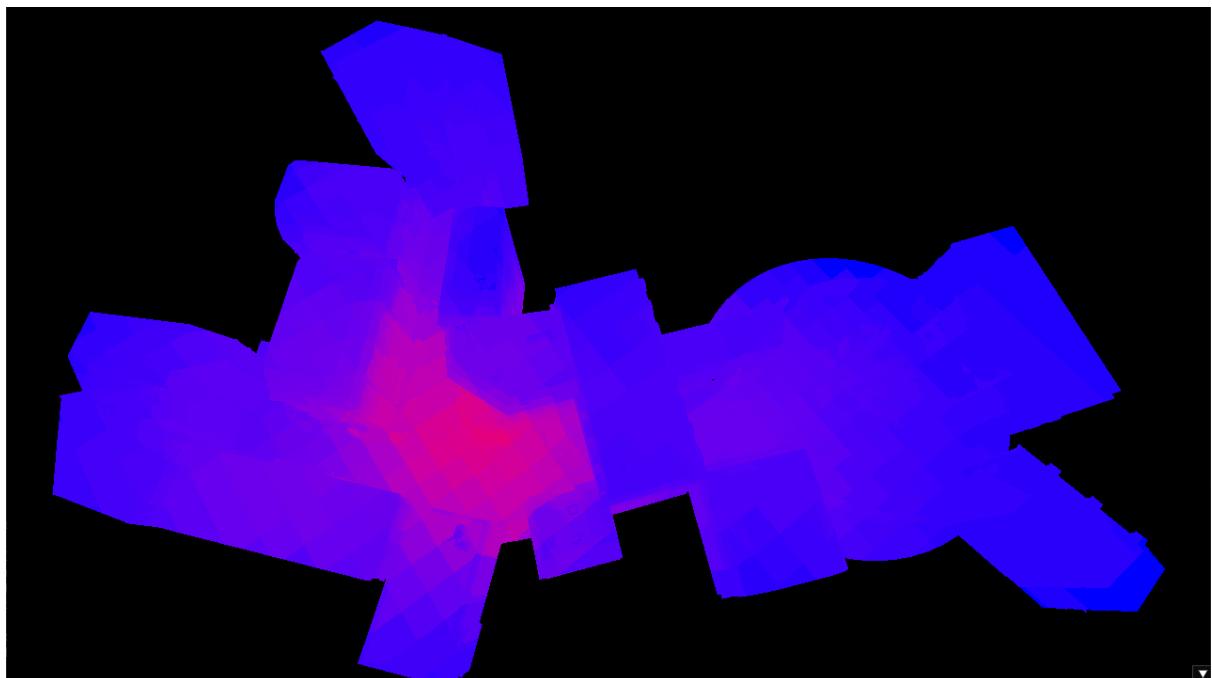
Nous avons considéré l'intensité lumineuse d'une source proportionnelle à la racine carrée de son aire totale.



Découpage de la scène en $8 \times 8 \times 8$ cellules.

Pour le light culling, la scène est découpée en une grille régulière de $32 \times 32 \times 32$ (= 32'768) cellules sur son domaine.

La liste des indices des sources lumineuses contribuant à chaque cellule est calculée. On obtient 188'222 indices, soit 5.74 sources lumineuses par cellules.



Concentration des sources lumineuses: en bleu, peu de sources lumineuses; en rouge, beaucoup.

Les sources lumineuses sont concentrées vers le milieu de la scène. Le nombre maximal de sources lumineuses contribuant à une même cellule s'élève à 83.

Une source lumineuse est considérée comme contribuant à une cellule si sa contribution maximale à la cellule est supérieure à un certain seuil.

Cette méthode est néanmoins limitée. L'atténuation par rapport à la vue permettrait de mieux évaluer la contribution d'une source à une cellule et ainsi de retirer plus de sources. La perception de la contribution d'une source dépend de la luminosité totale de la source; en effet, la perception lumineuse est logarithmique. Les sources contribuant faiblement pourraient être conservées si la luminosité totale est faible, et retirée, inversement.

En outre, le light culling bénéficierait d'être évalué à chaque image et pourrait être accéléré sur GPU.

Z-Prepass

Une Z-prépasse activable est fournie pour réduire ultimement le nombre d'évaluations de fragments.

Elle gère également la transparence pleine pour ne pas générer d'artefacts lors de l'illumination.



Depth buffer provenant de la Z-prépasse.

Rendu

Illumination

L'illumination est calculée avec le modèle de Blinn-Phong et ne conserve pas l'énergie.

Les valeurs de rugosité sont converties en valeurs de shininess pour être exploitées par ce modèle.



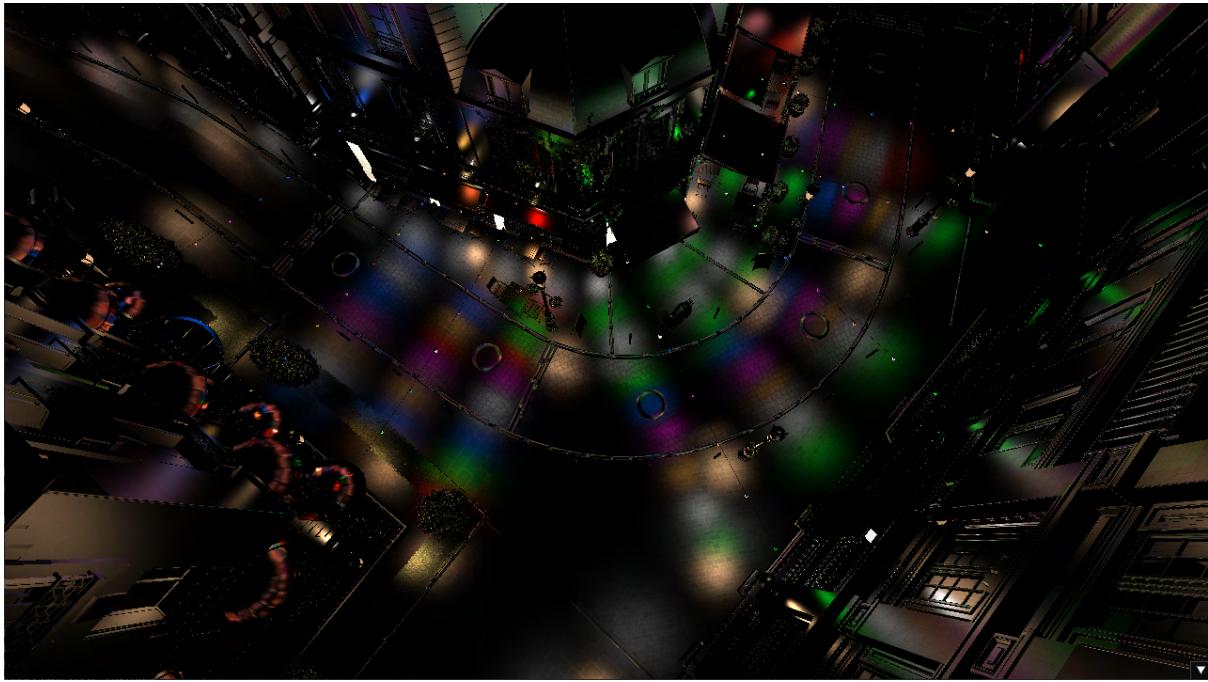
À gauche, le terme diffus. À droite, le terme spéculaire. En bas, l'illumination finale.

Pour chaque lumière contribuant à un fragment les termes diffus et spéculaires sont sommés. Un terme de réflectance lambertienne est calculé pour chaque lumière par rapport à la normale du fragment, et aussi pour la direction de la vue par rapport à la normale du fragment.



Terme lambertien de la direction de la vue par rapport au fragment.

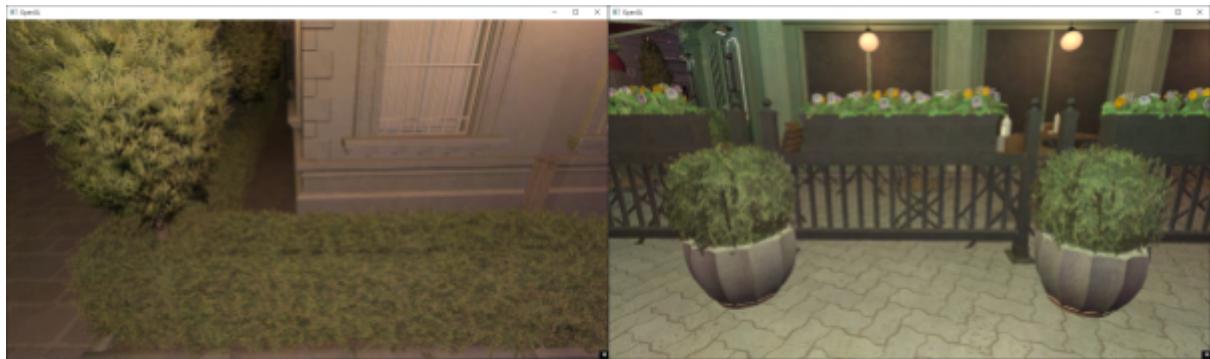
La scène ne comporte pas de lumière ambiante. Le terme spéculaire ne montre pas de reflets intenses du fait que la plupart des matériaux soient très rugueux.



Terme spéculaire en multipliant la shininess par 200 et l'intensité lumineuse par 100.

Une atténuation est calculée de la lumière au fragment puis du fragment à la lumière. La formule utilisée est $i / (d^3 + 1)$ avec i l'intensité lumineuse et d la distance.

Transparence



Implémentation de la transparence avec un alpha cutoff (0.5).

L'alpha cutoff doit également être implémenté dans la Z-prépassage pour ne pas comporter d'artéfacts. Il a donc fallu personnaliser cette passe pour y avoir accès à la texture albedo, comportant la valeur d'opacité.

Correction gamma



À gauche, sans; à droite, avec.

La correction gamma permet d'obtenir un meilleur contraste sur la scène et est visuellement bien plus plaisante.

Résultats



35 objets dessinés; temps moyen CPU: 1.2ms; temps moyen GPU: 10.1ms.



44 objets dessinés; temps moyen CPU: 1.4ms; temps moyen GPU: 5.5ms.



166 objets dessinés; temps moyen CPU: 1.3ms; temps moyen GPU: 8.7ms.



181 objets dessinés; temps moyen CPU: 1.2ms; temps moyen GPU: 10.3ms.



64 objets dessinés; temps moyen CPU: 1.3ms; temps moyen GPU: 16.1ms.