

UE-INF2317M Synthèse D'Image 3D – TP3



Université Claude Bernard



Lyon 1

COTTIER Alexandre – le code sur mon GitHub :

<https://github.com/Mantadoro1/synthese-d-image-3D-TP3>

1. Lancer de rayon, intersection avec triangle :

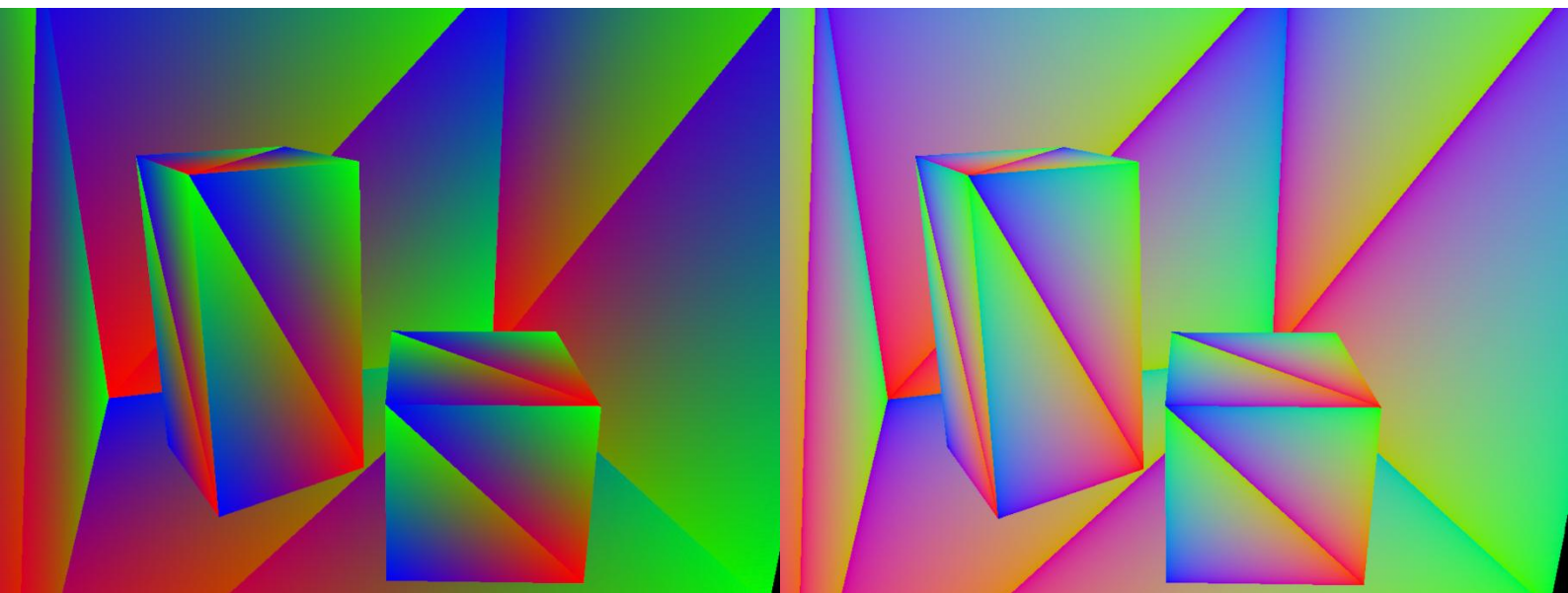
Pour chaque pixel (x,y) lance un rayon caméra : o = point (near), e = point (far) en inversant viewport * projection * view.

J'utilise l'algo de Moller-Trumbore, pour chaque triangle, je calcule s'il y a une intersection et je garde la plus proche (le t le plus petit et positif), j'ai donc les coordonnées barycentriques (u,v) du point touché.

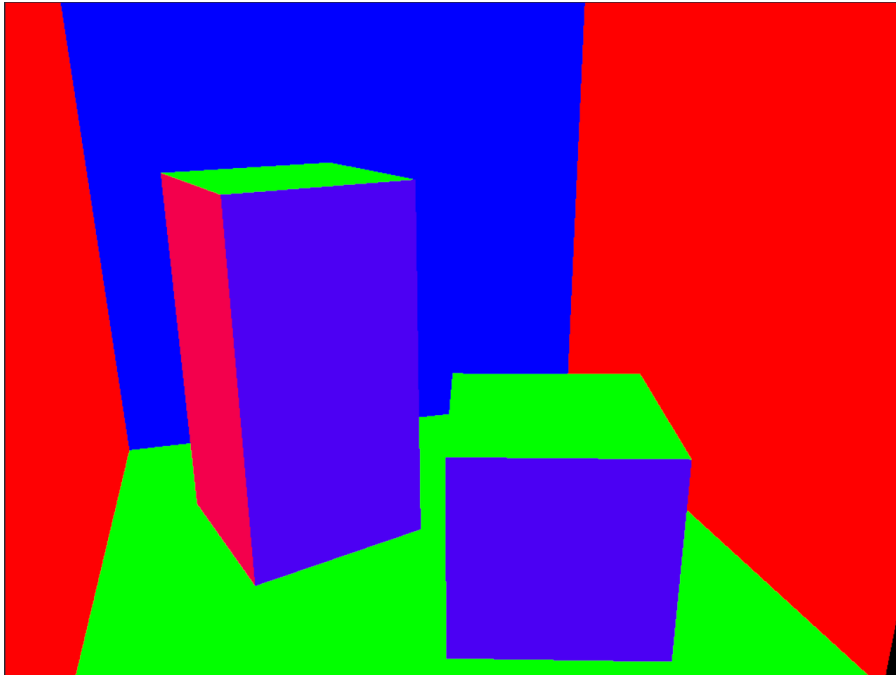
Je fais un petit visuel avec les barycentres (j'affiche la couleur $(1-u-v, u, v)$, juste parce que c'est intéressant et que dans d'autres contexte que ce tp je pense que ça peut être utile pour vérifier l'intersection) :

Figure 1 : version PNG

Figure 2 : version HDR



Idem ensuite pour les normales (j'interpole la normale $n = w \cdot n_a + u \cdot n_b + v \cdot n_c$ puis j'affiche $\text{abs}(n)$, dans un cas où on voudrait vérifier l'orientation) :

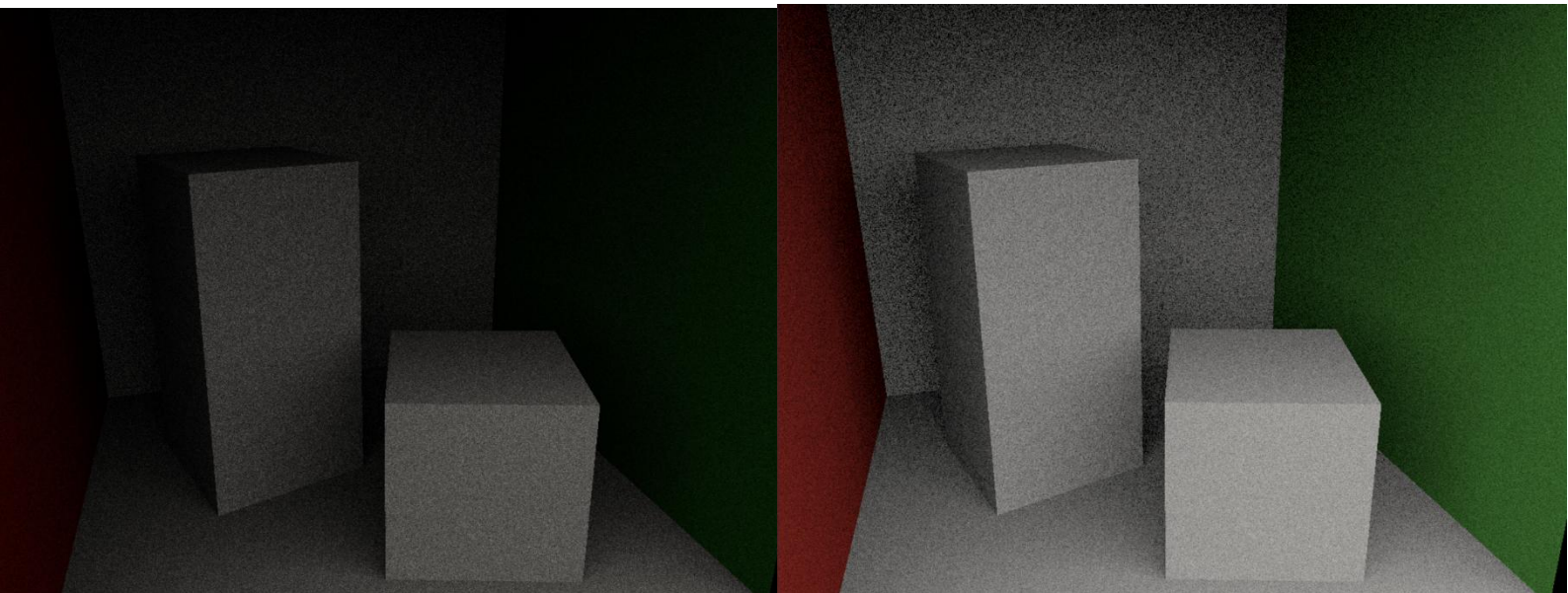


2. Lumière (Monte-Carlo)

Je modélise un ciel blanc ($L_i = 1$) et j'estime l'intégrale par monte-carlo, je tire N directions dans l'hémisphère, $(c/\pi) \cos \theta / \text{pdf}$ et je teste la visibilité vers le ciel, on a du coup les ombres portées et un rendu éclairé par la lumière du haut.

Puis du coup au lieu de $L_i = 1$ venant du ciel, je prends L_i depuis la direction du triangle touché, si le rayon secondaire touche un triangle avec $\text{material.emission} > 0$ et qu'on voit la face avant, alors j'utilise sa couleur d'émission, sinon 0.

Echantillon $N_{\text{sky}} = 64$

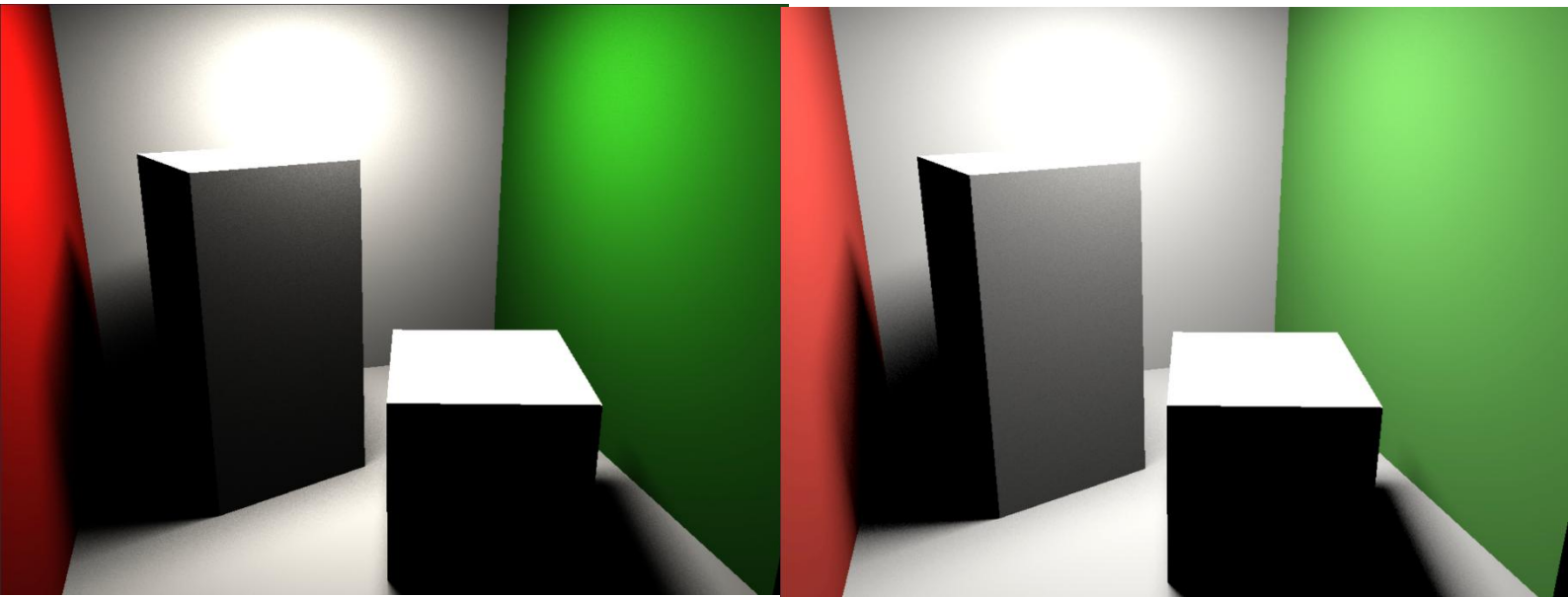


3. Eclairage direct efficace :

Ici je vise directement la lampe au lieu d'espérer la trouver avec des directions random.

Je construis une liste des triangles émissifs avec leur aire et la somme totale, ensuite pour calculer la contribution d'un point p, je tire un point q sur un triangle lumineux, proportionnel à l'aire, je calcule la géométrie et j'ajoute la contribution, je moyenne sur N points et on obtient du coup pas mal en qualité, on a des ombres nettes et moins de grain qu'avec les directions uniformes.

$$N_{dir} = 256$$



4. Eclairage indirect (1 rebond diffus)

Ici j'ajoute la lumière réfléchiée par les objets (color bleeding). La lumière incidente à p depuis une direction l vaut la lumière réfléchiée depuis le point y frappé dans cette direction.

Avec 1 rebond :

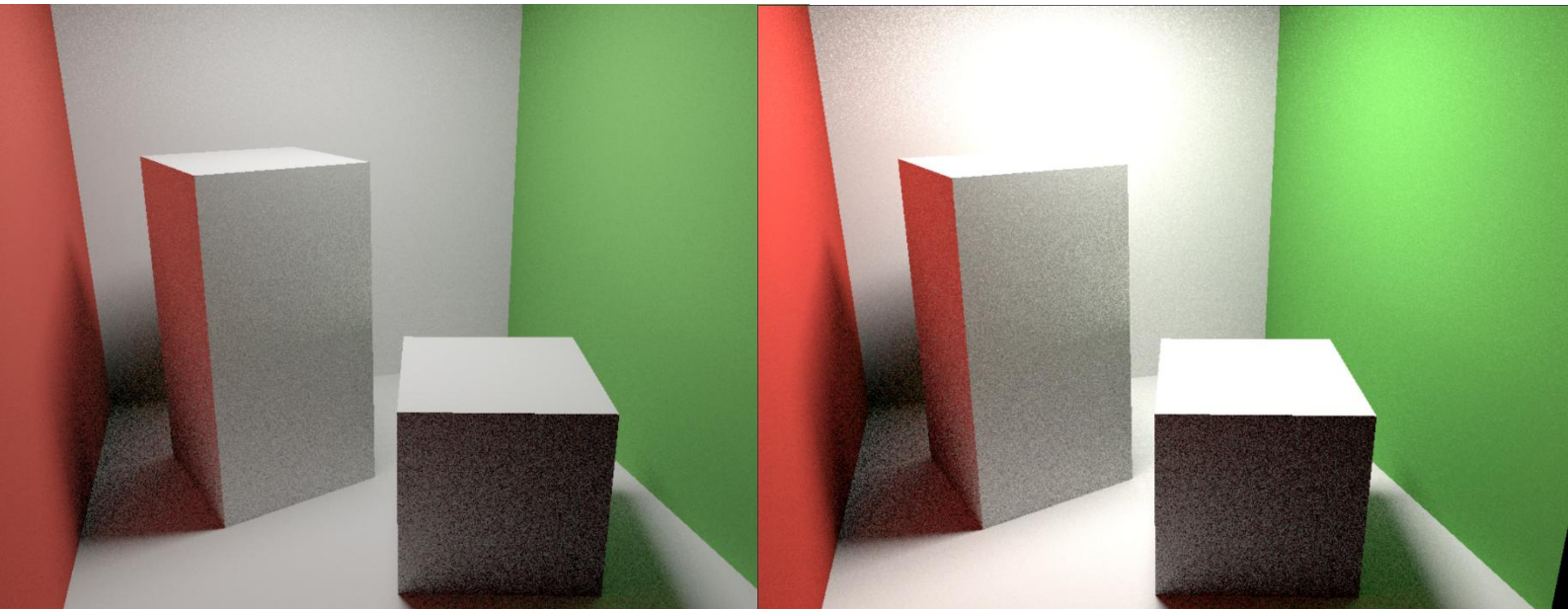
Au point p, je tire N_{ind} directions pondérées au cosinus

J'intersecte \rightarrow j'obtiens un point y sur un objet lumineux.

A y, j'évalue l'éclairage direct avec un seul échantillon d'aire

Je fais la contribution et au final j'ajoute $albedo(p) * Direct(y)$ et je moyenne

Ça nous donne la teinte des murs qui se propagent, un peu de grain mais qui diminue (on peut prendre N dir élevé 256, 512 etc)



Ici on prend $N_{dir} = 256$ et $N_{ind} = 16$

Pour les commandes, pour faire en sorte qu'on puisse bien voir chaque partie, j'ai fait en sorte qu'on puisse juste faire :

```
./bin/test data/cornell.obj data/cornell_orbiter.txt 0
```

```
./bin/test data/cornell.obj data/cornell_orbiter.txt 1
```

```
./bin/test data/cornell.obj data/cornell_orbiter.txt 2
```

```
./bin/test data/cornell.obj data/cornell_orbiter.txt 3
```

```
./bin/test data/cornell.obj data/cornell_orbiter.txt 4
```

0 étant la figure avec les normales,

1 étant la figure avec les barycentres,

2 étant l'éclairage direct,

3 étant l'éclairage direct par émetteur,

4 étant l'éclairage indirect.