

IFT6095 - Sujets en infographie

Travail pratique numéro 2 - Ombres interactives, lumières étendues
et environnementales.

21/02/2014

Anis Benyoub

1 Introduction :

Comme sources externes j'ai utilisé la librairie glmatrix pour la gestion des matrices et webgl-utils pour la boucle de rendu.

2 Ombres dures :

J'ai choisi d'implémenter le shadow mapping. Les choix d'implémentation que j'ai fait sont les suivant :

- Les calculs se font en espace image de la lumière
- On génère deux shadowmaps par lumière, chacune couvrant 90 degrés du champ de vision ce qui donne deux hémisphères.

Pour chaque rendu de la scène on rend dans un premier temps la scene depuis le point de vue de la lumière, on effectue un deuxième rendu depuis la même position mais en effectuant une rotation de 90° autour de l'axe de rotation supérieur.

Le résultat de ces rendus est stocké dans une texture et ce grâce à un framebuffer. Étant donné que les depth textures ne sont pas disponibles par défaut dans la version courante de webgl, il a fallu les stocker dans des textures de couleur en prenant bien soin d'homogénéiser la profondeur et de les packer en utilisant des fractales pour les bits de poids faible.

On effectue ensuite un deuxième rendu en bindant les deux textures ainsi que la view matrix de la lumière afin de pouvoir effectuer les tests. On teste ensuite si le fragment dans les deux espaces image de la lumière sont dans l'espace clipping et si c'est le cas on teste la différence de profondeur (après avoir unpack les données stockées dans la texture de profondeur).

Pour l'intégration d'une primitive plus complexe, j'ai implémenté un importeur d'objets de type .obj. La lumière est mobile et se déplace de manière sinusoïdale dans la scène.

Voici un visuel de rendu avec deux scènes différentes :



FIGURE 1 – *Rendu de deux scenes*

Les artefacts que j'ai rencontré lors de mon implémentation de shadow mapping sont les suivants : Decalage en profondeur(il peut être résolu en rajoutant un decalage lors de la comparaison). Un autre problème est le manque de précision du tampon de profondeur (accru a cause de la taille UNSIGNED_BYTE imposé par webgl).



FIGURE 2 – *Mise en évidence du problème de decalage en profondeur et precision du tampon de profondeur*

Le troisième est l'aliassage spacial :



FIGURE 3 – *Mise en evidence du probleme d'alissage spacial*

3 Lumières polygonales :

Pour cette partie, j'ai implémenté la partie obligatoire mais aussi la partie bonus.

Pour la partie obligatoire, le rendu se fait en 3 temps, un peu comme du deferred shading. On rends les polygone une fois par lumière et en changeant la fonction de blending on somme la contribution de chaque lumière. Idéalement pour faire du deferred shading il aurait fallut faire les calcul en esopace image, mais les limitations du webgl font que j'ai préféré ne pas faire ça avec un G-Buffer

La principale difficulté que j'ai rencontré dans cette partie a été le probleme de clipping des polygones et ce dû aux limitations de webgl (impossibilité de faire un accès à un tableau avec un indice non constant). L'algorithme utilisé pour cette partie m'est propre.

En ce qui concerne l'algorithme d'éclairage je n'ai pas utilisé l'implémentation proposée par les articles, je me suis uniquement servi des formules mathématiques.

Voici le rendu visuel final :

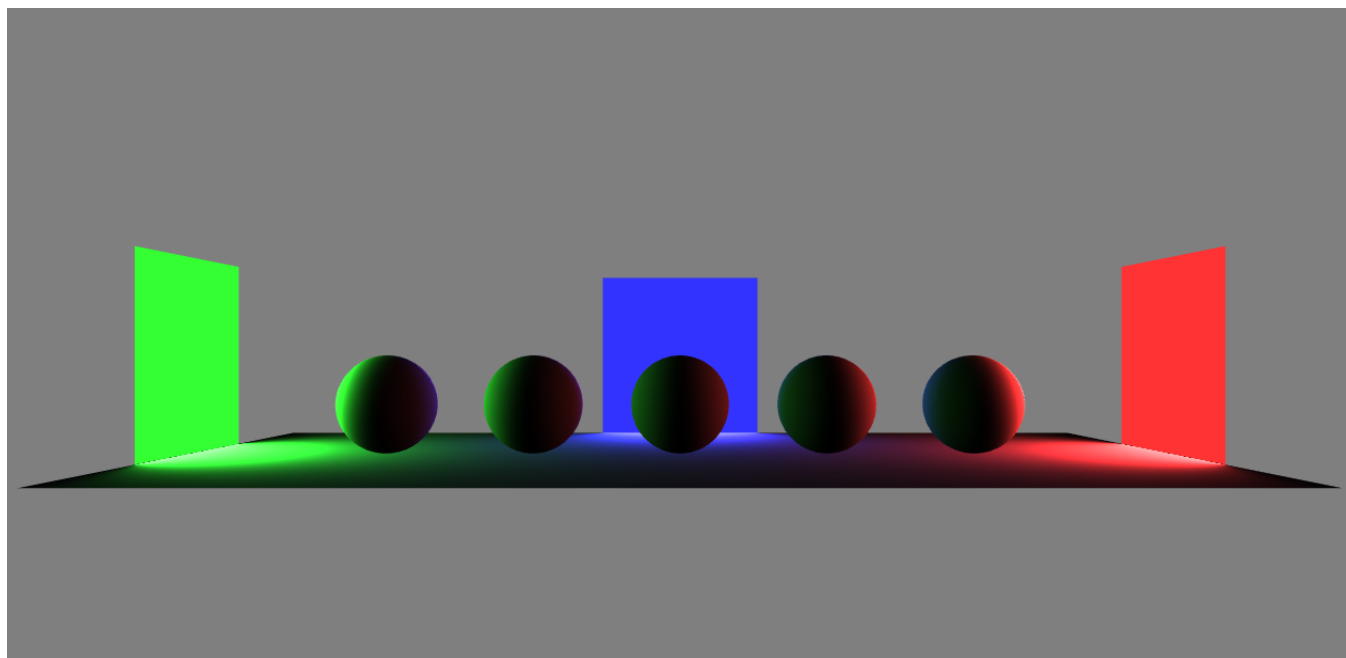


FIGURE 4 – *Rendu de la scene avec BRDF diffuse*

Concernant la partie optionnelle je n'ai pas utilisé l'implémentation proposée par l'article de recherche. Je me suis uniquement servi des formules mathématiques. J'ai effectué le clipping autour de la normale et non pas du vecteur réfléchi car si on se place derière la

lumière et le fragment il n'y a plus d'éclairage.

Il y a je pense encore un bug d'implémentation qui fait que l'on voit des sphère lumineuses là où il est censé y avoir. Voici le rendu final obtenu :

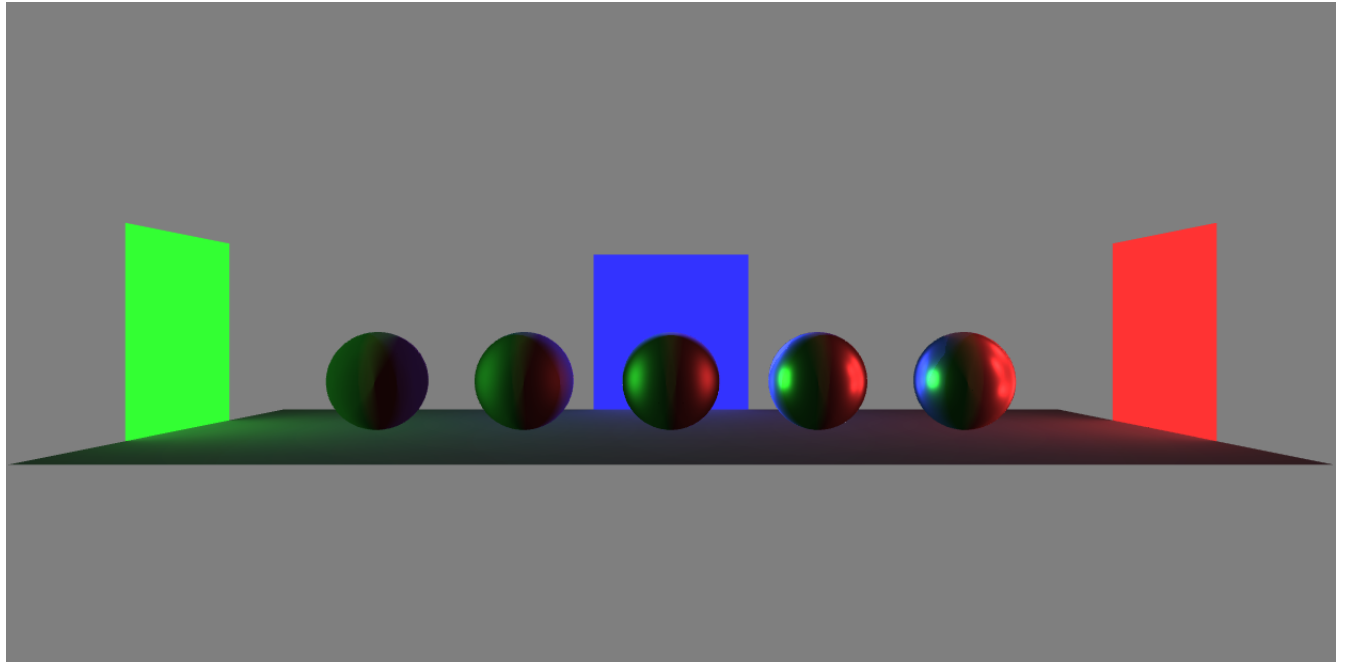


FIGURE 5 – *Rendu de la scene avec BRDF Phong*

Deux problèmes sont survenus lors de l'implémentation de cet algorithme. Le premier est dû au fait de ne pas pouvoir faire de boucle à indice final variable, j'ai été obligé de faire des boucles à indice maximal avec un break quand la boucle doit se terminer.

Le deuxième est l'impossibilité d'implémenter la récursivité, ce problème se résout assez facilement en faisant de manière itérative.

4 Lumière environnementale :

Dans cette partie, j'ai fait uniquement la partie concernant l'implémentation de l'article de Rammamoorthi et Hanrahan. Il suffit d'appliquer l'équation N°13 pour chaque composante de chaque fragment. J'injecte en uniform trois tableaux de 9 coefficients représentant les coefficients de chacune des harmoniques sphériques.

Voici le rendu final de cette partie :



FIGURE 6 – *Rendu de la scene avec lumière environnementale pour chacune des 3 lumières*

5 Remarque personnelle :

Ce TP était très intéressant, malheureusement il est très difficile de trouver le temps de faire toutes les parties proposée. Je pense que je vais essayer de les faire de mon coté quand j'aurais un peu plus de temps.