



ÉCOLE
CENTRALELYON

ÉCOLE CENTRALE LYON

RAPPORT

MOS Informatique graphique Rapport - Ray tracing

Élèves :
Ayoub TOUDGHI

Enseignant :
Nicolas BONNEEL

Table des matières

1	Introduction :	2
2	Construction de la scène :	2
2.1	Eclairage lambertien, ombres portées et Correction gamma :	2
2.2	Surfaces miroirs et surfaces diffuses :	2
2.3	Eclairage indirect avec source étendue et ombres douces :	3
2.4	Depth of field :	4
2.5	Maillages triangles avec Bounding Box :	4
2.6	Textures	5
2.7	Repo Github	5

1 Introduction :

Dans le cadre de notre cours d'informatique graphique, le présent rapport détaille les différentes étapes suivies pour parvenir au rendu final de notre travail sur le ray tracing. Le ray tracing est une technique de rendu photoréaliste qui simule le comportement des rayons de lumière dans une scène tridimensionnelle, permettant ainsi de générer des images de haute qualité. Ce rapport présentera en détail les différentes phases de développement, en mettant en avant les résultats obtenus à chaque étape. Des éléments d'implémentation clés seront également discutés pour une meilleure compréhension du processus de création.

Les résultats sont créées avec une résolution de 512x512. L'éclairage est fixé à 2 milliards de lux, cependant, les autres variables sont modifiées à chaque itération pour améliorer la visualisation. Le temps d'exécution est enregistré pour chaque résultat.

2 Construction de la scène :

2.1 Eclairage lambertien, ombres portées et Correction gamma :

Pour la première étape de mon projet, j'ai construit une scène composée de différentes sphères. Ensuite, j'ai calculé l'intersection entre un rayon et une sphère pour placer les différents composants de la scène. Par la suite, j'ai utilisé le modèle d'éclairage lambertien, en considérant une source d'éclairage ponctuelle d'intensité I . De plus, j'ai pris en compte le calcul des ombres en ajoutant un terme de visibilité.

J'ai également ajouté une correction gamma pour la visibilité des faibles intensités.

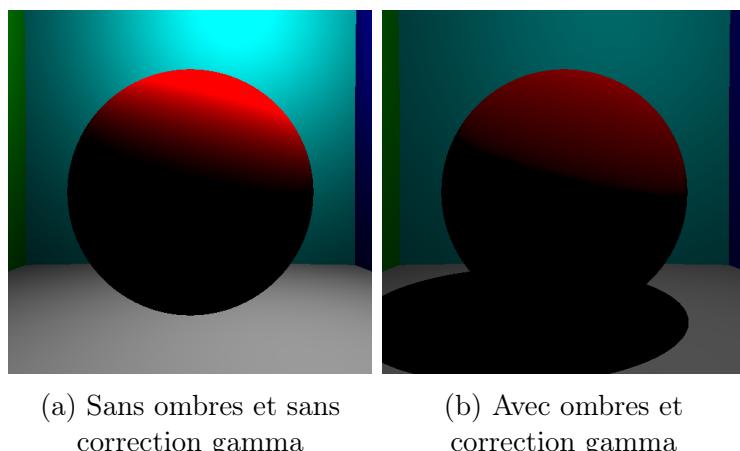


FIGURE 1 – Eclairage lambertien, ombres portées et Correction gamma

2.2 Surfaces miroirs et surfaces diffuses :

j'ai également exploré l'éclairage des surfaces miroir et des surfaces diffuses. Les surfaces miroir, contrairement aux surfaces diffuses, réfléchissent la lumière de manière spéculaire, c'est-à-dire qu'elles produisent des reflets nets et précis. Pour modéliser cet effet, j'ai implémenté le calcul de la réflexion spéculaire en utilisant la loi de réflexion de la lumière. Cette loi stipule que l'angle d'incidence d'un rayon lumineux est égal à l'angle de réflexion par rapport à la normale à la surface.

D'autre part, les surfaces diffuses diffusent la lumière dans toutes les directions, produisant des réflexions douces et uniformes.

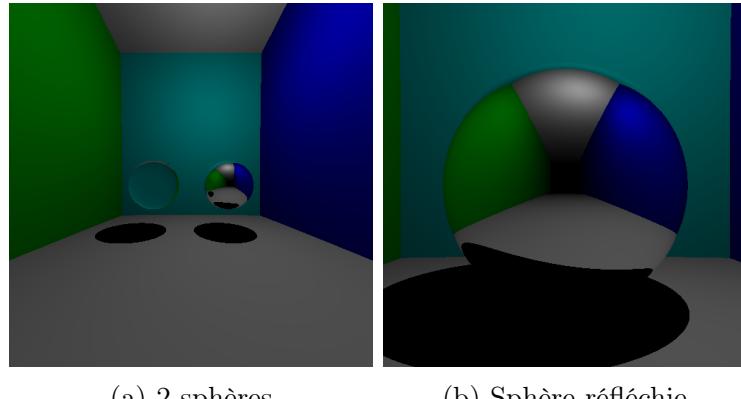


FIGURE 2 – Surfaces miroirs et surfaces diffuses

2.3 Eclairage indirect avec source etendue et ombres douces :

Dans la modélisation de la lumière pour les scènes rendues par raytracing, l'éclairage indirect est essentiel, j'ai adopté une approche de rendu global pour calculer cet éclairage de manière récursive, en utilisant la fonction de distribution de la reflectance bidirectionnelle (BRDF) des surfaces. J'ai également élargi ma méthode pour inclure une source de lumière étendue avec le calcul des ombres douces, améliorant ainsi la qualité visuelle des rendus grâce à des transitions graduées entre les zones éclairées et ombragées.

De plus, j'ai ajouté une étape d' **anti-aliasing** pour réduire les artefacts visuels tels que l'aliasing et les crênelages dans mes rendus, permettant ainsi d'améliorer la qualité globale des images en adoucissant les transitions et en rendant les contours plus lisses.

Enfin, les résultats sont présentés ci-joint, démontrant l'impact de l'utilisation de différents nombres de rayons envoyés pour éliminer le bruit et améliorer la qualité des rendus avec le temps d'exécution Te et avec I=2E9 et une source de lumière étendue de rayon R = 15.

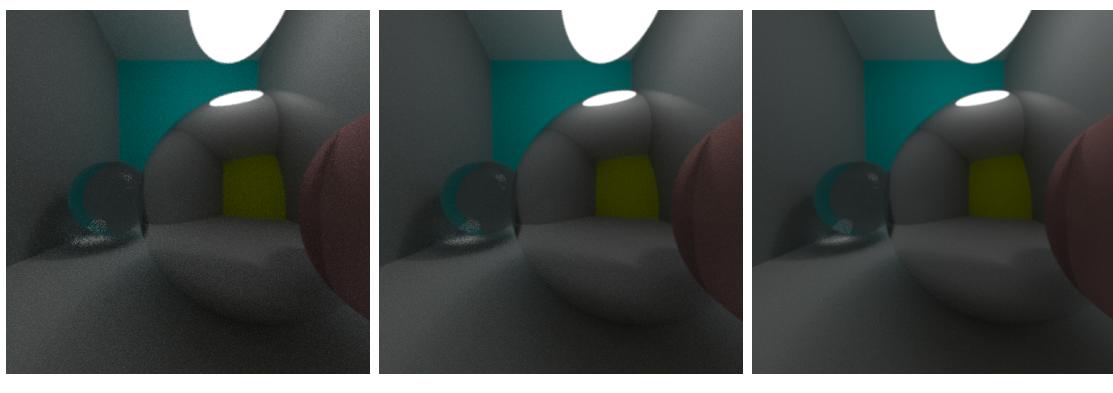


FIGURE 3 – Eclairage indirect avec source etendue et ombres douces

2.4 Depth of field :

Pour simuler des effets réalistes tels que la profondeur de champ et le flou de mouvement dans mes rendus, j'ai évité d'utiliser le modèle sténopé (pinhole) en raison de ses limitations. À la place, j'ai opté pour l'utilisation d'un diaphragme pour contrôler la quantité de lumière entrant dans la caméra, offrant ainsi une plus grande flexibilité pour la création d'effets de flou réalistes. En intégrant ces effets, j'ai pu améliorer la qualité visuelle de mes rendus et créer des scènes plus immersives.

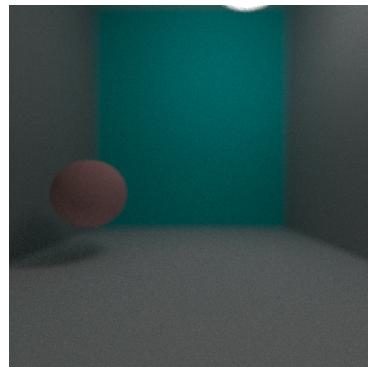


FIGURE 4 – Depth of field nryas = 30, Te = 183s

2.5 Maillages triangles avec Bounding Box :

J'ai d'abord étendu mon système pour inclure des objets avec des maillages triangulaires, ce qui m'a permis de calculer les intersections avec ces triangles à travers des bounding boxes. Cependant, cette approche s'est avérée chronophage, nécessitant beaucoup de calculs. Par exemple, la génération des bounding boxes pour une scène donnée a pris 430 secondes pour un nombre de rayon = 100 (fig 6a) et avec un rescale = 0.3 et translation. Afin d'améliorer l'efficacité de ce processus, j'ai opté pour une stratégie hiérarchique et continue. J'ai pu diminuer de temps de calcul à 201s avec nryas = 100 comme illustre ci joint dans fig 6b.

De plus, j'ai ajouté le lissage de Phong en moyennant la normale par les coefficients calculés lors de l'intersection, ce qui a permis d'améliorer la qualité visuelle des rendus (fig 6c).

Les résultats sont inclus ci-joint. Il m'a fallu un certain temps pour les générer en fonction des caractéristiques de lumière utilisées et du nombre de rayons. Pour une meilleure visibilité, vous pouvez consulter les résultats sur le dépôt GitHub, dans le dossier "résultats".

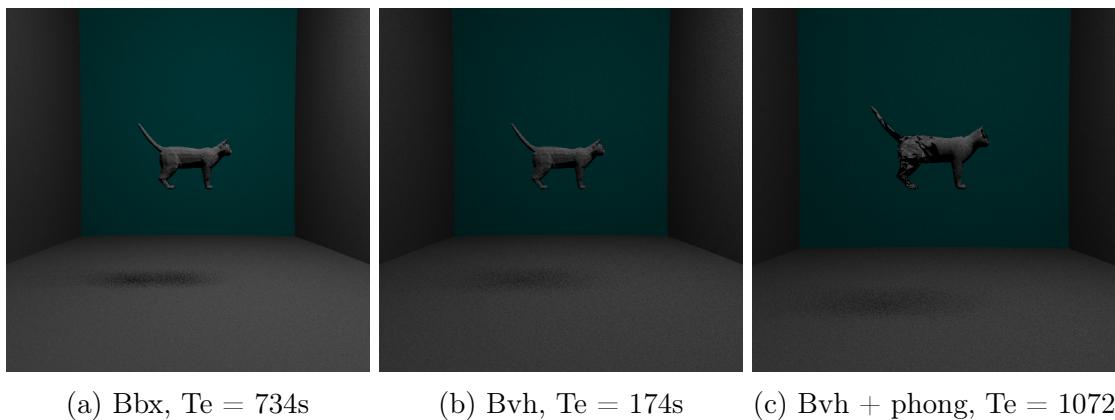


FIGURE 5 – Resultats du Maillage nrays = 10

2.6 Textures

J'ai intégré la lecture et l'affichage de la texture. J'ai également étendu les fonctionnalités en intégrant des méthodes pour mettre à l'échelle, translater et faire pivoter des objets dans l'espace. De plus, j'ai incorporé une fonctionnalité permettant d'appliquer un type spécifique de texture aux sphères de la scène.

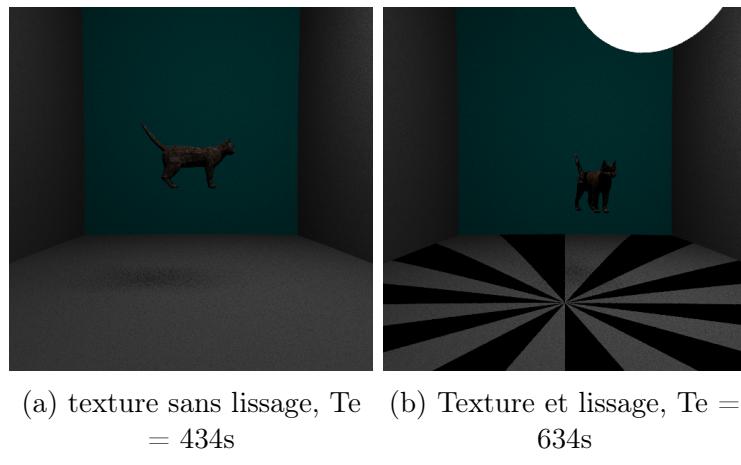


FIGURE 6 – Resultats du Maillage avec texture nrays = 10

2.7 Repo Github

Dans le fichier main.cpp, vous trouverez tout le code utilisé, tandis que le dossier "Résultats" contient toutes les sorties pour chaque étape nécessaire à la création de la scène.