



# ÉCOLE CENTRALE LYON

## INFORMATIQUE GRAPHIQUE RAPPORT

# Résultats des différentes techniques de rendu des images en utilisant la méthode du Ray Tracing

*Élèves :*

Camilo GARCÍA VERGARA

*Enseignant :*

Nicolas BONNEEL

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Principe</b>	<b>2</b>
<b>3</b>	<b>Lumière ponctuelle, éclairage direct et sphères</b>	<b>3</b>
<b>4</b>	<b>Ombres, transparences, surfaces miroir et correction gamma</b>	<b>5</b>
<b>5</b>	<b>Éclairage indirect</b>	<b>9</b>
<b>6</b>	<b>Lumières étendues, anti-aliasing, et profondeur de champ</b>	<b>13</b>
<b>7</b>	<b>Maillages, lissage de Phong et filtre median</b>	<b>17</b>
<b>8</b>	<b>Textures, BRDF de Phong et réflexion Fresnel</b>	<b>22</b>
<b>9</b>	<b>Conclusion</b>	<b>25</b>

## 1 Introduction

Ce rapport rend compte du travail réalisé lors des séances de BE et d'autonomie du cours Informatique Graphique.

Pendant le cours, nous avons travaillé avec le Ray Tracing, une méthode de rendu d'images qui consiste à simuler le parcours inverse de la lumière pour décrire l'éclairage des objets dans une scène. Nous avons vu la gestion des différents types de surfaces, soit miroir, soit des transparences, soit des matériaux diffus, et même l'utilisation des textures. Si bien dans le cours nous avons vu plusieurs contenus, il est vrai que la totalité n'a pas été réussie. Alors, dans ce rapport nous montrerons la partie réussie des contenus, ainsi que les différents problèmes que nous avons trouvé au cours de la réalisation des algorithmes développés.

## 2 Principe

À la base, pour travailler avec la méthode du Ray Tracing nous avons besoin d'au moins les éléments suivants :

- La caméra
- Un écran
- Une source de lumière
- Des éléments dans la scène

Dans la réalité, les sources de lumière envoient des rayons qui arrivent vers les objets. Ces objets absorbent une certaine partie de ces rayons, et le reste est réfléchi et donc visible par une caméra. Dans le Ray Tracing, nous faisons le chemin à l'inverse : étant donné un écran de  $W \times H$  pixels, nous envoyons des rayons depuis la caméra vers chaque pixel et nous essayons de trouver des intersections avec des objets de la scène. Si nous trouvons une intersection, nous envoyons un rayon dans la direction de la lumière pour pouvoir calculer l'intensité de l'éclairage du point d'intersection. La Figure 1 montre un schéma simple de ce principe avec les composants décrits. Dans cette Figure, nous pouvons voir également la gestion des ombres. Nous parlerons de cela plus tard.

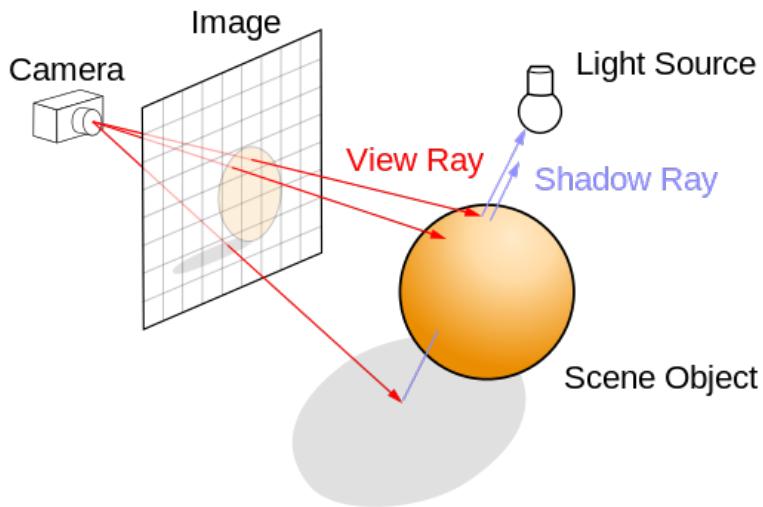


FIGURE 1 – Sch ma simple du Ray Tracing.

Source: [fr.wikipedia.org/wiki/Ray\\_tracing](http://fr.wikipedia.org/wiki/Ray_tracing)

### 3 Lumi re ponctuelle,  clairage direct et sph res

Dans une premi re partie, nous avons travaill  avec une source de lumi re ponctuelle. Le principe a  t  assez simple : nous avons une sc ne avec des sph res d'une certaine couleur et nous envoyons un rayon par pixel de notre image. La couleur est d finie par un vecteur de trois dimensions, une pour chaque canal rouge, vert et bleu. Les parois sont cr  es en utilisant des sph res de tr s grand rayon pour simuler des plans.

Dans les Figures 2 et 3 nous pouvons voir le type de r sultats que cette premi re approche fournit. Des d tails   remarquer : tout d'abord, il est visible que les centres des parois sont plus claires  clair s que les bords. Cela est  vidence de la proximit  des centres   la lumi re (les parois sont des sph res). En plus, nous remarquons que les sph res du centre de la sc ne sont  clair es selon leur position par rapport   la lumi re. Ainsi, dans la Figure 2 la sph re du centre est presque noir, parce que la lumi re est plus loin de la cam ra. Par contre, dans la Figure 3 la sph re   droite est plus loin que la lumi re, donc elle est allum e par rapport   la position de la cam ra.

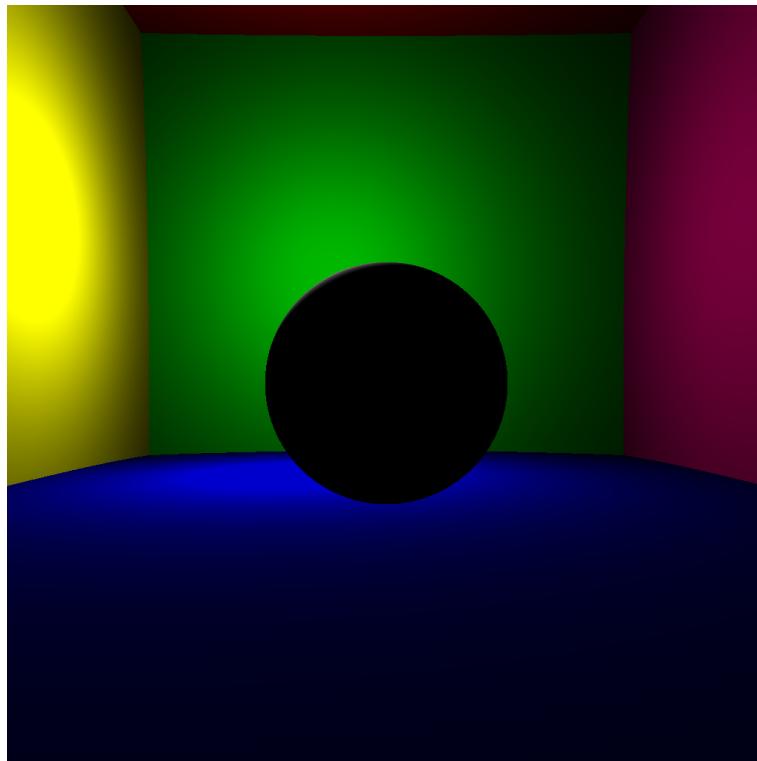


FIGURE 2 – Premier exemple d'éclairage ponctuel

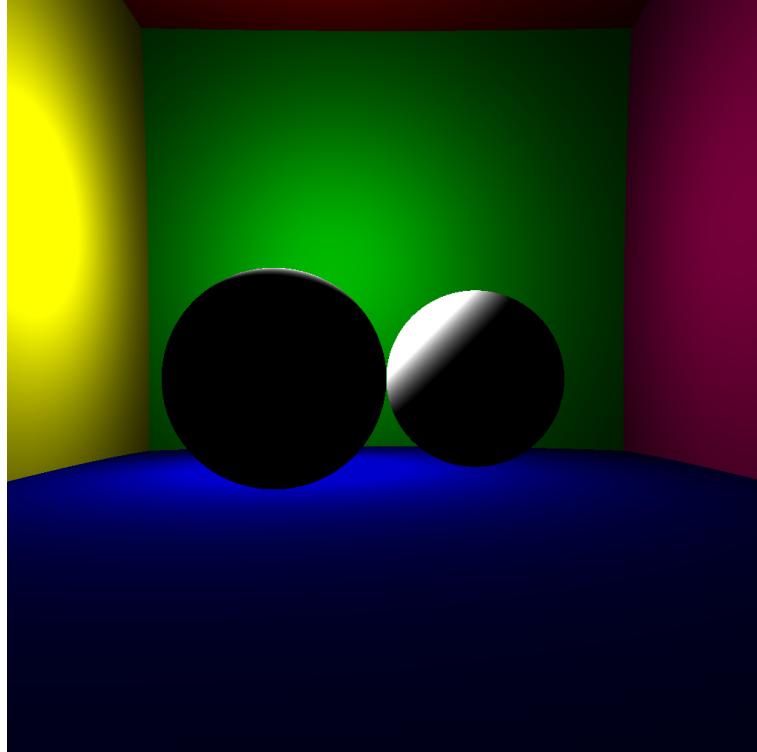


FIGURE 3 – Deuxième exemple d'éclairage ponctuel

## 4 Ombres, transparences, surfaces miroir et correction gamma

Dans la réalité, nous avons des matériaux différents. En effet, les objets que nous voyons n'ont pas seulement des surfaces diffuses, sinon aussi des surfaces purement spéculaires et des objets transparents. En plus, la présence d'un objet implique la production d'une ombre. Si un objet **A** est entre un objet **B** et la source de lumière, l'objet **A** ne permet pas à la lumière d'éclairer l'objet **B**. Alors, l'objet **A** produit une ombre sur l'objet **B**.

En ce qui concerne les ombres, la première approche est assez simple : pour chaque rayon envoyé depuis la caméra, si nous trouvons une intersection, nous vérifions que le chemin entre le point d'intersection et la source de lumière ponctuelle soit libre. Si c'est le cas, le point d'intersection est allumé. Cependant, s'il y a un objet que s'interpose entre la lumière et le point d'intersection, ce point d'intersection sera noir (avec une ombre). Nous pouvons voir le résultat de cette gestion des ombres dans la Figure 4. Effectivement, les parois et le sol ont des parties noires dues aux sphères qui empêchent le passage de la lumière.

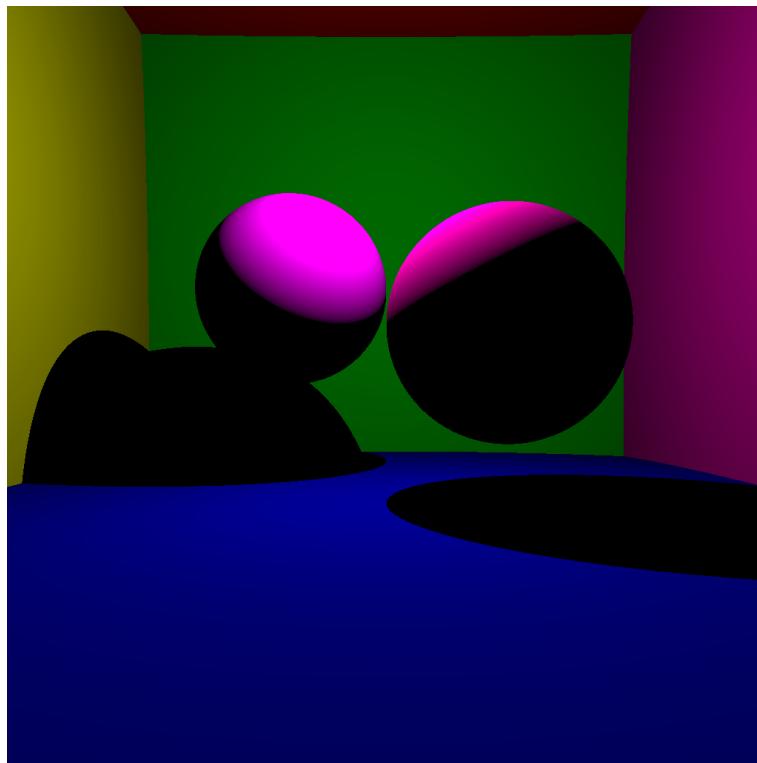


FIGURE 4 – Crédit : <https://www.cs.cmu.edu/~patterson/courses/15-461/lectures/shadow.pdf>

Les surfaces miroir sont celles qui réfléchissent la totalité de la lumière incidente. En plus, cette réflexion est faite avec un angle égal à l'angle du rayon de lumière incident à la surface par rapport à la normale dans le point de réflexion. La Figure 5 montre le schéma des rayons incident **I** et réfléchi **R** par rapport à la normale **N**.

Alors, pour obtenir la couleur d'une surface miroir, il faut vérifier le point d'arrivée du rayon réfléchi. En fait, ce rayon réfléchi pourrait éventuellement tomber sur une autre surface miroir et générer un deuxième rayon réfléchi. Cette situation pourrait arriver plusieurs fois et ne va finir que quand un des rayons réfléchis soit tombé sur une surface

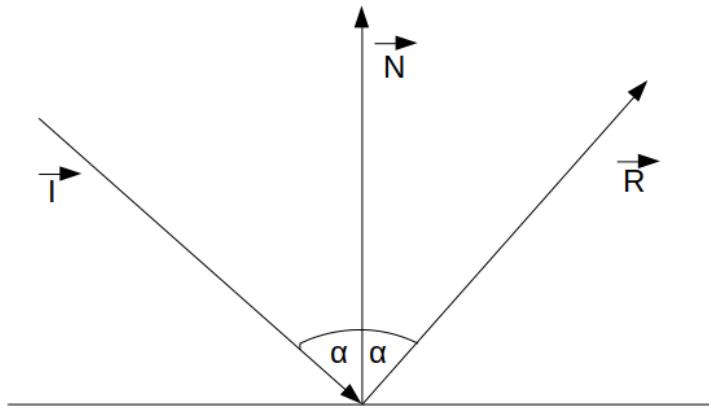


FIGURE 5 – Schéma du rayon réfléchi dans une surface miroir

diffuse. En effet, dans certains cas il pourrait arriver que deux surfaces miroir génèrent des rayons parallèles qui vont être réfléchis de manière infinie. Pour éviter cela, nous avons limité la quantité de rebonds à 5 maximum. Nous pouvons voir les résultats obtenus dans la Figure 6. Il est visible dans cette Figure que les rebondissements répétés semblent d'être bien gérés.

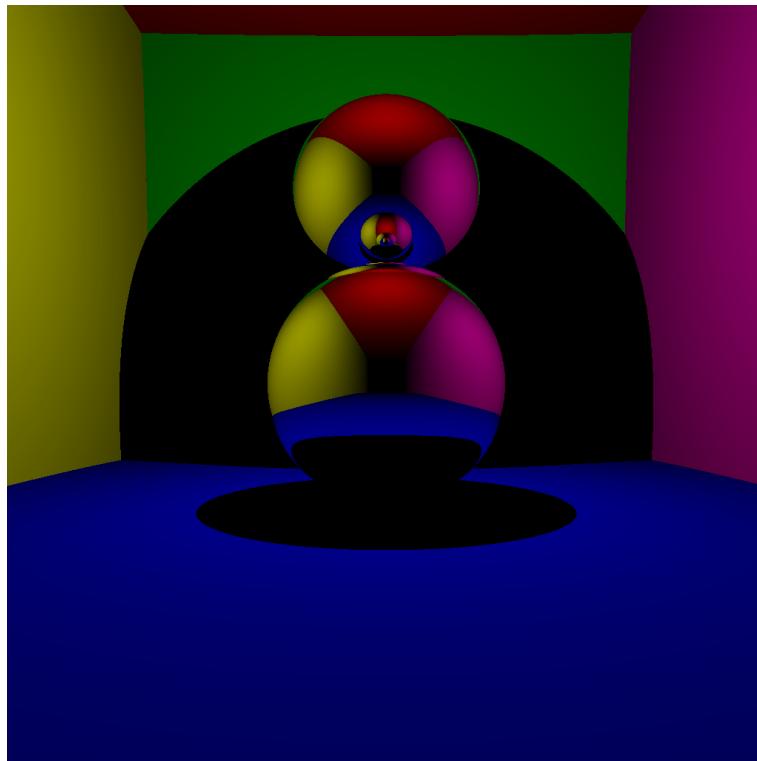


FIGURE 6 – Résultat de la gestion des surfaces miroir

En relation aux surfaces transparentes, nous les avons géré en suivant la loi de Snell selon le schéma décrit dans la Figure 7 :

$$n_1 \sin \sigma_1 = n_2 \sin \sigma_2$$

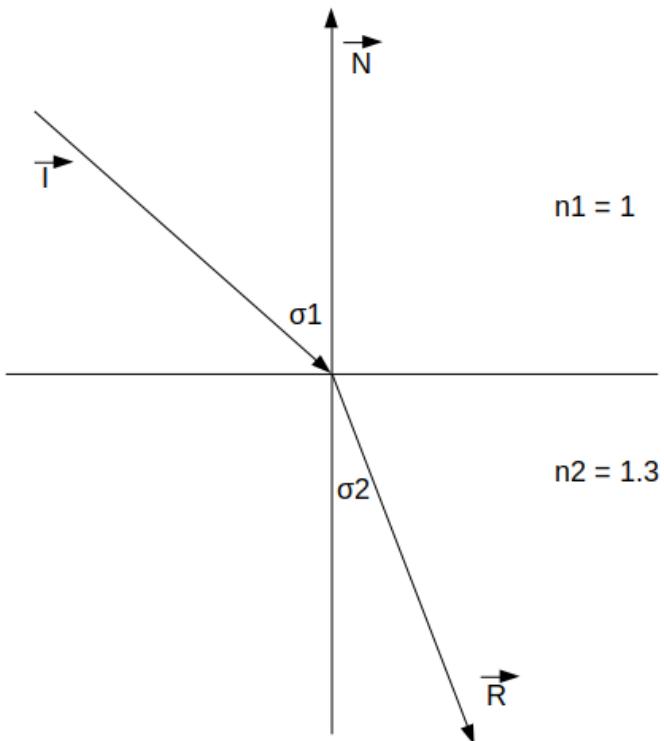


FIGURE 7 – Schéma des surfaces transparentes. Nous avons utilisé des indices de réfraction  $n_1 = 1$  pour le milieu et  $n_2 = 1.3$  pour les sphères transparentes

Compte tenu qu'il y a un rayon réfracté chaque fois que la lumière change de milieu (entrée et sortie d'une sphère transparente), nous avons le même problème que dans les surfaces spéculaires : la quantité de rayons pourrait être trop élevée. Alors, l'algorithme développé pour obtenir la couleur d'un point de la scène serait récursif dans les cas spéculaire et transparent, et aura une limite de 5 nouveaux rayons (soit réfractés, soit réfléchis). Dans la Figure 8 nous pouvons voir une image avec une sphère d'un matériau transparent avec un indice de réfraction de 1.3. Effectivement, nous pouvons voir l'inversion des couleurs des parois au moment de traverser la sphère transparente.

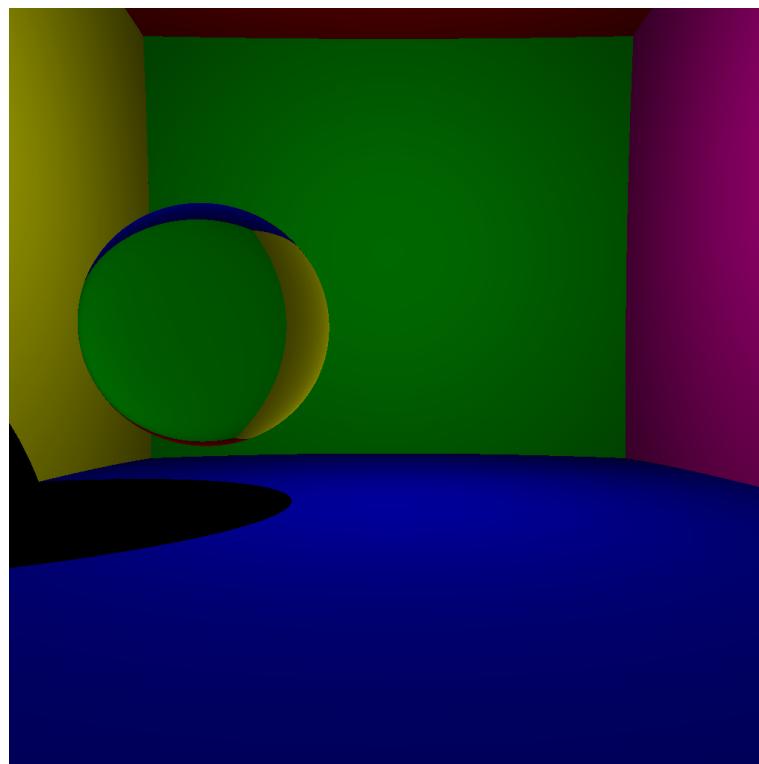
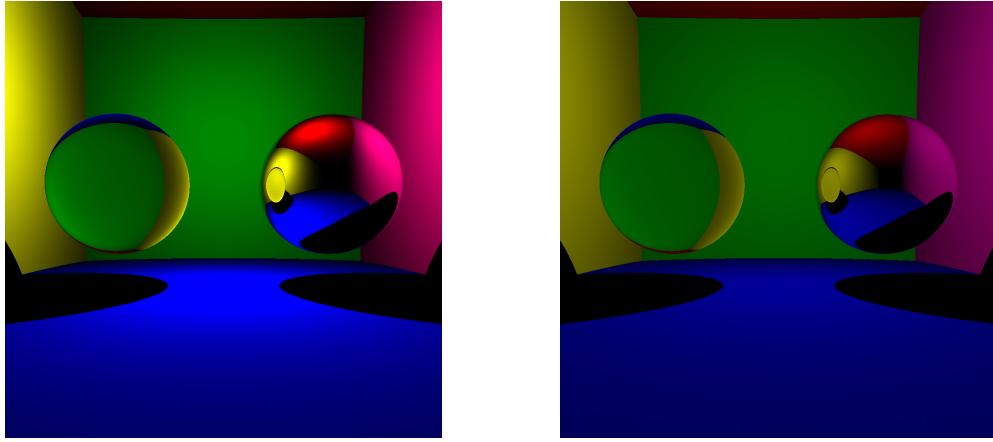


FIGURE 8 – Résultat de la gestion des surfaces transparentes

Dernièrement, nous avons appliqué une correction gamma aux images. Il s'agit d'un facteur permettant d'améliorer le contraste d'une image sans perdre les détails dans les ombres, alors une non-linéarité dans la distribution des couleurs des pixels pour compenser les différences de sensibilité des yeux humains. Alors, pour chaque pixel et sa couleur originale  $pixel_{in}$ , nous avons généré une valeur corrigée  $pixel_{out}$  selon la formule :

$$pixel_{out} = pixel_{in}^{\frac{1}{2.2}}$$

Nous pouvons voir l'effet de la correction gamma dans la Figure 9. En effet, les pixels où la lumière arrive de manière plus puissante sont moins brillants après la correction gamma. De la même façon, les pixels qui étaient plus ombrés sont maintenant plus clairs.



(a) Image sans la correction gamma avec une intensité de lumière ponctuelle de 500000

(b) Image avec la correction gamma avec une intensité de lumière ponctuelle de 100000000

FIGURE 9 – Application de la correction gamma. Il faut remarquer que, dû à la puissance  $\frac{1}{2.2}$ , l'intensité de la lumière doit varier pour garder un aspect proche entre les deux images

## 5 Éclairage indirect

Un des principaux défauts des images précédentes est la gestion des ombres. L'approche que nous avons utilisé est trop simple, parce que nous donnons directement la couleur noir aux points de la scène dont la lumière est cachée par un autre objet. Cependant, dans la réalité les objets ne sont pas éclaires uniquement par les sources de lumière, sinon aussi par les objets diffus qui réfléchissent une certaine partie de la lumière qu'ils reçoivent. Pour générer une approche plus réaliste, nous avons utilisé le concept de la BRDF, de l'anglais Bidirectional reflectance distribution function. Le principe est le suivant : étant donné un rayon incident dans une surface, il peut être réfléchi dans toutes les directions de l'hémisphère de la surface où il arrive avec une certaine probabilité. À titre d'exemple, dans la Figure 10 nous voyons que le rayon incident  $\Omega_i$  peut être réfléchi avec une probabilité non nulle dans un cône décrit par  $d\Omega_r$ .

Nous introduisons alors l'équation du rendu :

$$L_o(\vec{x}, \vec{\Omega}_r, \lambda) = E(\vec{x}, \vec{\Omega}_r, \lambda) + \int_{S^2} f(\vec{x}, \vec{\Omega}_i, \vec{\Omega}_r, \lambda) L_i(\vec{x}, \vec{\Omega}_i, \lambda) \cos \sigma_i d\vec{\Omega}_i$$

où :

- $E$  est l'émissivité de la surface
- $f$  est la BRDF
- $L_o$  est l'intensité lumineuse qui en sort
- $L_i$  est l'intensité lumineuse qui arrive au point
- $\vec{x}$  est le point d'arrivée du rayon
- $\sigma_i$  est l'angle entre  $\Omega_i$  et la normale à la surface

Compte tenu que  $L_i$  d'un point dépend de  $L_o$  d'autres points, nous avons une intégrale récursive pour calculer  $L_o$ .

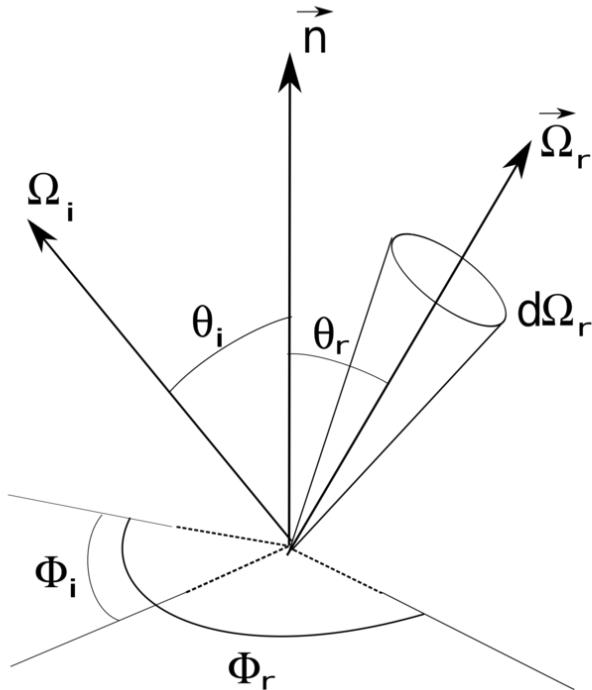


FIGURE 10 – Principe d'une BRDF.

**Source:** [fr.wikipedia.org/wiki/Réflectivité\\_bidirectionnelle](https://fr.wikipedia.org/wiki/Réflectivité_bidirectionnelle)

Pour pouvoir calculer l'intégrale, nous avons utilisé la méthode de Monte-Carlo. En plus, du fait que nous utilisons une source de lumière ponctuelle, nous n'avons pas des surfaces émissives, donc il faut simuler l'éclairage directe. Finalement, si nous prenons la BRDF  $\frac{\rho}{\pi}$  des surfaces diffuses, ainsi que la fonction de probabilité  $p(\vec{\Omega}_i) = \frac{\cos \sigma_i}{\pi}$  pour la méthode de Monte-Carlo, nous arrivons à la formule suivante pour  $L_o$  :

$$L_o = \frac{1}{N} \sum L_i \rho + I \frac{\rho}{\pi} \frac{(\vec{L} - \vec{x}) \cdot \vec{N}}{d^2}$$

où :

- $N$  est la quantité de rayons envoyés selon la méthode de Monte-Carlo
- $\rho$  est l'albédo de l'objet intersecté par le rayon
- $\vec{L}$  est le vecteur depuis le point d'intersection vers la source de lumière
- $d$  est la distance entre le point d'intersection et la source de lumière

Dans les Figures 11 et 12 nous pouvons voir le résultat d'envoyer 100 rayons ( $N = 100$ ). Effectivement, les ombres deviennent plus doux et nous pouvons voir la couleur des parois réfléchi sur les objets diffus du centre de la scène.

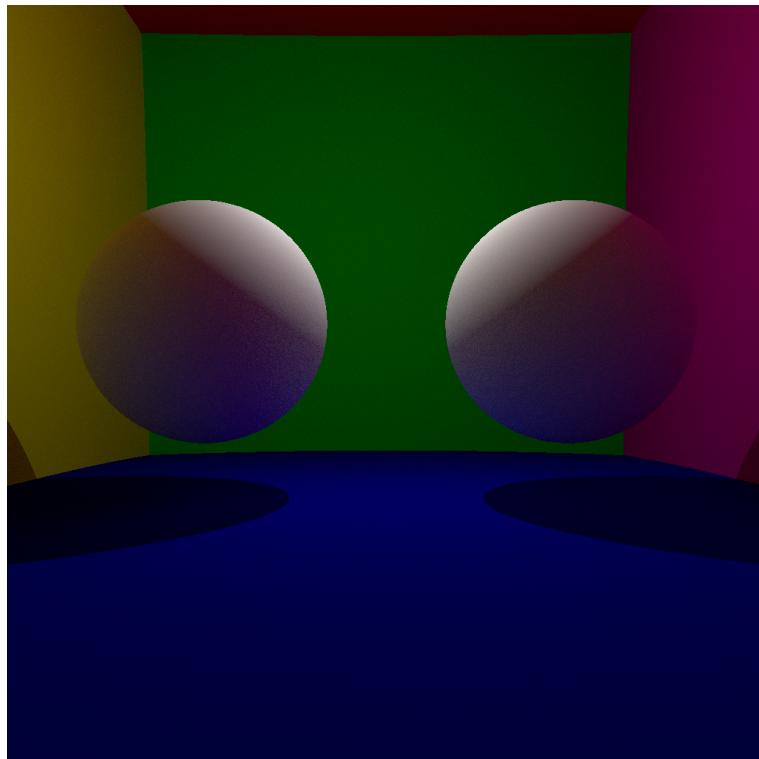


FIGURE 11 – Résultat de la gestion de l'éclairage indirect

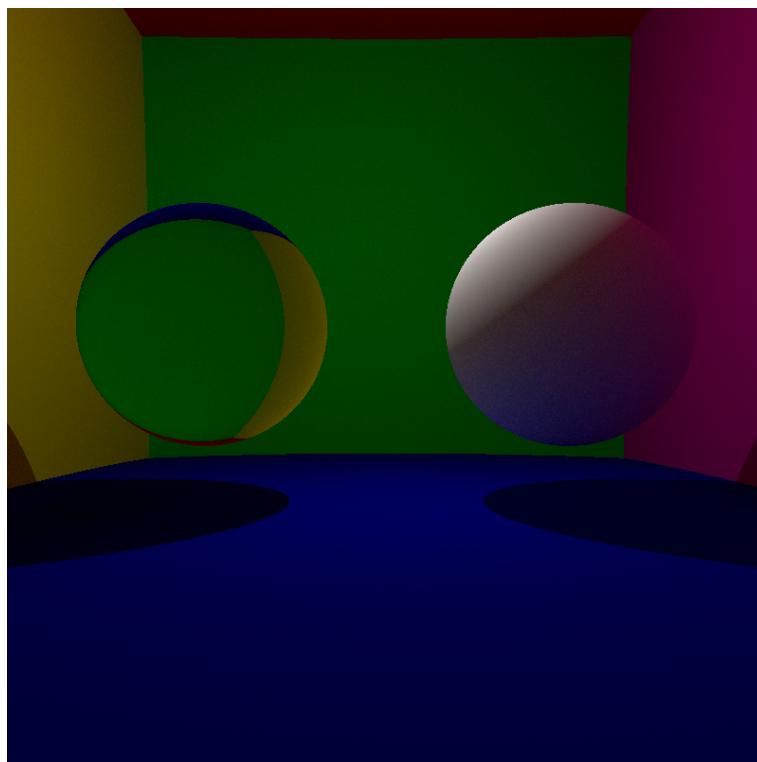


FIGURE 12 – Résultat de la gestion de l'éclairage indirect avec une sphère transparente

Étant donné que la méthode de Monte-Carlo est une approximation de l'intégrale réelle, nous trouvons certain niveau de bruit dans nos résultats. La Figure 13 montre le bruit présent dans une image de taille 1024x1024 dont nous avons utilisé 100 rayons par pixel. Augmenter la quantité de rayons par pixel permet d'obtenir des images avec moins du bruit, cependant, le temps rendu de l'image peut devenir très élevé. En fait, le code pour obtenir les images des Figure 11 et 12 a eu un temps d'exécution de 10 minutes environ.

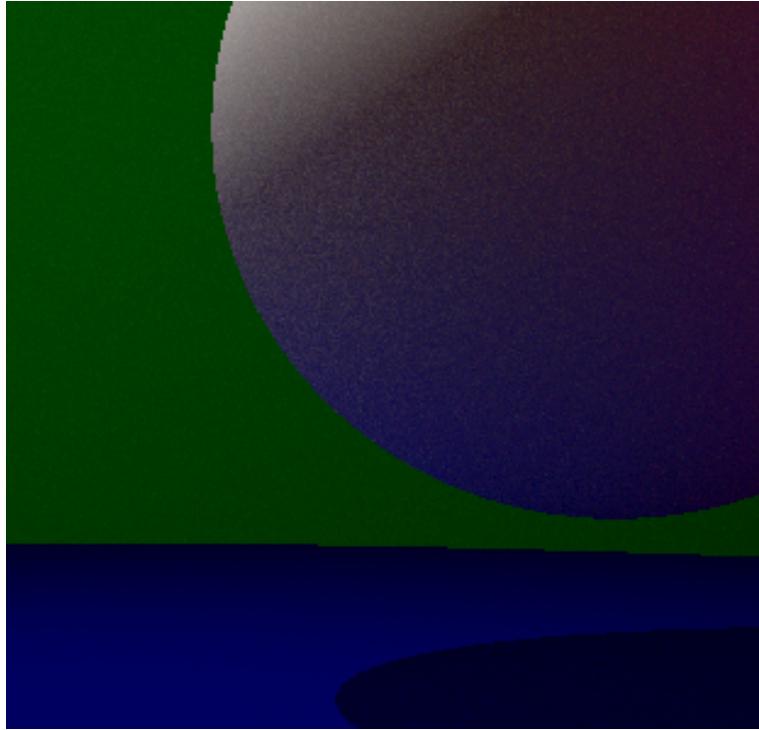


FIGURE 13 – Bruit dans les images dû à l’approximation de l’intégrale par la méthode de Monte-Carlo

## 6 Lumières étendues, anti-aliasing, et profondeur de champ

Normalement, les sources de lumière sont des objets avec un certain volume et pas tout simplement un point comme nous avons défini pour le moment. Dans le cas de ces objets, nous trouvons qu’ils ont un facteur d’émissivité non nul mais une BRDF nulle. Compte tenu que nous utiliserons maintenant une source de lumière sphérique, il faut prendre en compte que son intensité est distribuée dans toute sa surface. Alors, la formule utilisé pour déterminer l’intensité de la lumière est la suivante :

$$I = \frac{C}{4\pi R^2 \pi}$$

où  $C$  est une constante. En plus, pour faire le code plus rapide, nous avons fait un changement de variable : à la place d’envoyer des rayons aléatoires, nous générerons des points aléatoires dans la scène. Finalement, l’approximation de l’intégrale de l’équation de rendu par la méthode de Monte-Carlo après le changement de variable est la suivante :

$$\frac{1}{N} \sum \frac{\frac{I \rho}{\pi} \cos \sigma \frac{V(x, x_i) \cos \sigma'}{\|x - x_i\|^2}}{\frac{\cos \sigma''}{\pi R^2}}$$

où :

- $V(x, x_i)$  est la visibilité entre le point  $x$  et  $x_i$ ;
- $\sigma'$  est l’angle entre le rayon généré par les points  $x$  et  $x_i$  et la normale au point  $x_i$

- $\sigma''$  est l'angle entre le rayon généré par le point  $x$  et le centre de la sphère où nous avons généré le point  $x_i$ , et le rayon généré par ce centre et le point  $x$ ;

Le schéma proposé dans la Figure 14 sert pour montrer de manière plus claire ce qui vient d'être expliqué.

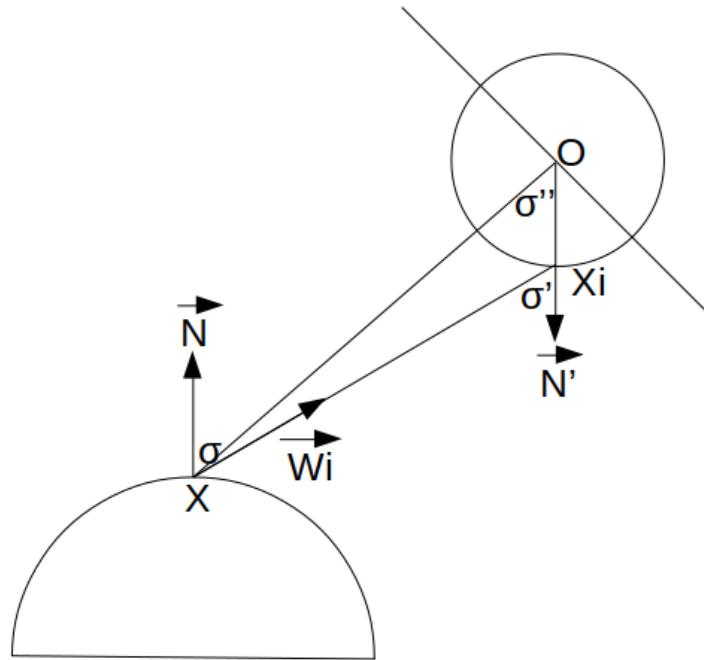


FIGURE 14 – Schéma pour montrer le détail de l'approximation de l'intégrale par la méthode de Monte-Carlo après le changement de variable

Nous pouvons voir les résultats de la gestion des lumières étendues dans la Figure 15. Nous avons mis en place la lumière dans un point de la scène de manière qu'elle soit visible. En effet, elle est représentée comme une sphère blanche.

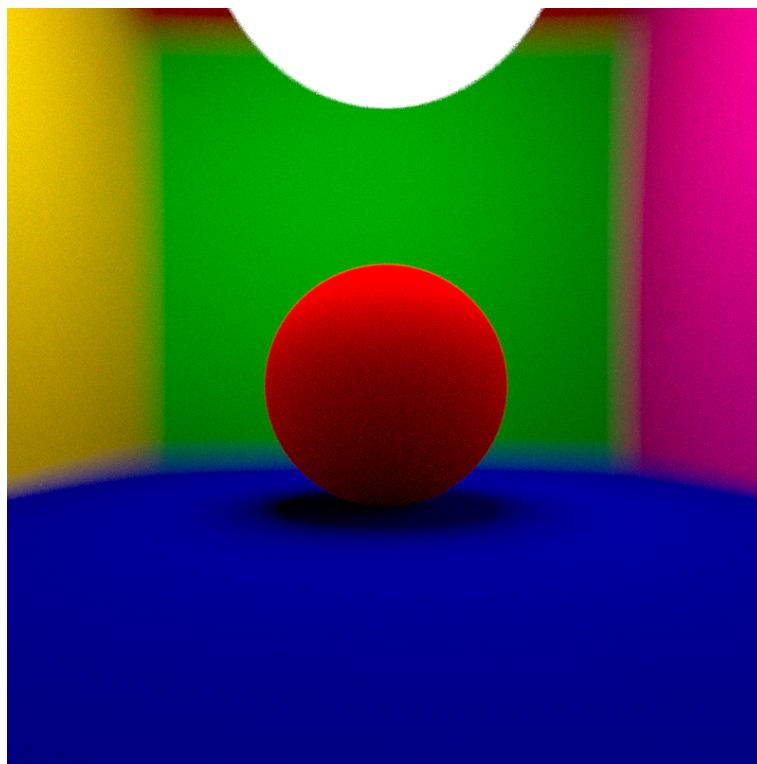
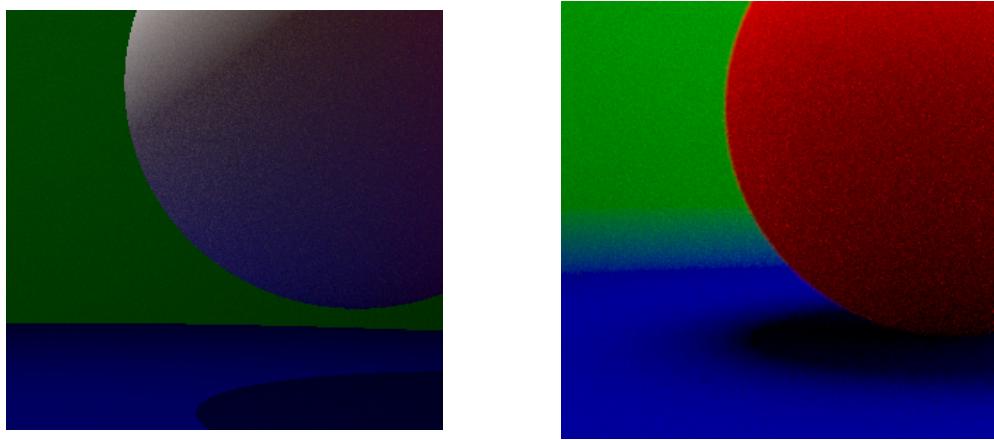


FIGURE 15 – Résultat de la gestion des sources de lumière étendues

Un autre problème que nous avons jusqu'à maintenant est le fait que les bords des sphères sont pixellisés : cela est dû au fait que nous envoyons toujours des rayons depuis la caméra jusqu'à un point spécifique du pixel de l'image (coin gauche en haut). La Figure 13 nous permet de voir cela. Pour éviter ce problème dans nos images, nous avons appliqué la méthode de Box-Muller. Cette méthode consiste à envoyer des rayons vers différents points du pixel et avec un certain poids. Nous allons donner plus de l'importance aux valeurs de couleur renvoyées par les rayons plus proches du centre du pixel. Toutefois, le reste aura de toute manière un poids non nul. La Figure 16 rend compte de l'amélioration obtenue par cette méthode : en effet, les bords sont plus doux même pour des images de taille inférieure.



(a) Image de 1024x1024 pixels sans la correction anti-aliasing

(b) Image de 800x800 pixels avec la correction anti-aliasing

FIGURE 16 – Utilisation de la méthode de Box-Muller pour obtenir des bords plus doux.

Nous remarquons que même pour une image plus petite, le bords sont plus doux que dans l'approche précédente

Finalement, nous avons traité les effets de profondeur de champ. Normalement, les lentilles des caméras ont une certaine distance focale, ce qui permet d'avoir un plan où les points sont nets dans l'image. Dans les caméras, cet effet est produit par le diaphragme : selon son ouverture les rayons lumineux vont générer des points plus ou moins flous pour les éléments qui ne sont pas dans le plan à la distance focale. Pour obtenir cet effet dans nos images, nous avons changé notre caméra : à la place de lancer des rayons depuis un point central (la caméra) comme nous avons fait jusqu'à maintenant, nous envoyons des rayons depuis des points aléatoires dans un carré centré dans le point de la caméra et suivant une distribution gaussienne. De cette manière, chaque point que ne soit pas dans le plan focal de la caméra deviendra flou. Dans la Figure 15 nous voyons déjà cet effet dans les parois. Cependant, la Figure 17 rend compte de la gestion du profondeur du champ dans une scène avec trois sphères centrales, chacune avec un des matériaux différent que nous avons géré jusqu'à maintenant. Il est visible également dans cette Figure que les ombres générées par les sphères transparents sont plus réalistes maintenant par rapport à ce que nous pouvons voir dans la Figure 12.

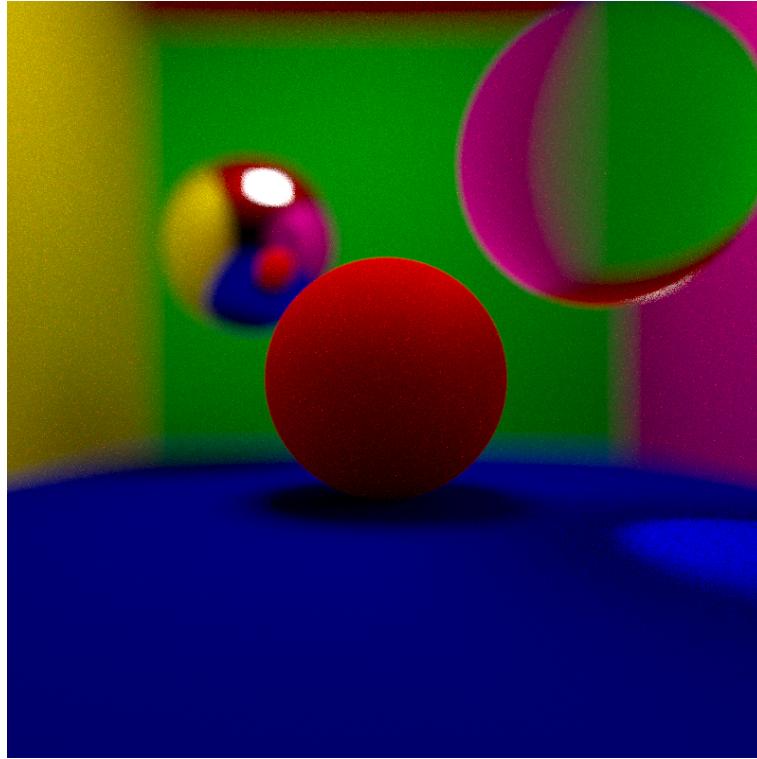


FIGURE 17 – Résultat de la gestion du profondeur de champ avec des sphères miroir, transparentes et diffuses

## 7 Maillages, lissage de Phong et filtre median

Pour le moment nous n'avons travaillé qu'avec des sphères, par contre cela n'est pas toujours la meilleure option. Compte tenu de cela, nous avons géré des intersections avec des plans, et plus particulièrement avec des triangles pour pouvoir obtenir finalement des géométries plus complexes.

Un plan peut être décrit par un point  $\mathbf{C}$  et sa normale  $\vec{N}$ . Ainsi, un point  $\mathbf{P}$  appartient au plan si :

$$(\mathbf{C} - \mathbf{P}) \cdot \vec{N} = 0$$

Alors, l'intersection d'un rayon  $R : \mathbf{P} = \mathbf{O} + t\vec{u}$  avec le plan précédent est donnée par :

$$t = \frac{(\mathbf{C} - \mathbf{O}) \cdot \vec{N}}{\vec{u} \cdot \vec{N}}$$

où  $\mathbf{O}$  est l'origine du rayon et  $\vec{u}$  sa direction. Nous nous intéressons aux valeurs de  $t > 0$ , car elles sont les intersections devant la caméra selon notre système de coordonnées. En plus, il faut toujours vérifier que  $\vec{u} \cdot \vec{N}$  soit différent de zéro.

Pour limiter cette intersection à un triangle, nous avons utilisé la méthode des coordonnées barycentriques. Étant donné un triangle  $ABC$ , nous avons décrit un point  $\mathbf{P} = \alpha A + \beta B + \gamma C$ , où  $\alpha$ ,  $\beta$  et  $\gamma$  sont les coordonnées barycentriques. Cela peut être écrit également avec des vecteurs vers  $A$  :  $\vec{AP} = \beta \vec{AB} + \gamma \vec{AC}$ . Avec cela, si nous faisons

des projections sur  $\overrightarrow{AB}$  et sur  $\overrightarrow{AC}$ , nous arrivons à un système d'équations linéaire. Par la méthode de Cramer, les inconnues  $\alpha$ ,  $\beta$  et  $\gamma$  sont données par :

$$\beta = \frac{\det \begin{pmatrix} \overrightarrow{AP} \cdot \overrightarrow{AB} & \overrightarrow{AC} \cdot \overrightarrow{AB} \\ \overrightarrow{AP} \cdot \overrightarrow{AC} & \|\overrightarrow{AC}\|^2 \end{pmatrix}}{\det(M)}$$

$$\gamma = \frac{\det \begin{pmatrix} \|\overrightarrow{AB}\|^2 & \overrightarrow{AP} \cdot \overrightarrow{AB} \\ \overrightarrow{AB} \cdot \overrightarrow{AC} & \overrightarrow{AP} \cdot \overrightarrow{AC} \end{pmatrix}}{\det(M)}$$

$$\alpha = 1 - \beta - \gamma$$

où la matrice M est égale à :

$$\begin{bmatrix} \|\overrightarrow{AB}\|^2 & \overrightarrow{AC} \cdot \overrightarrow{AB} \\ \overrightarrow{AB} \cdot \overrightarrow{AC} & \|\overrightarrow{AC}\|^2 \end{bmatrix}$$

Nous disons que le point  $\mathbf{P}$  appartient au triangle  $ABC$  si :

- $0 < \alpha, \beta, \gamma < 1$
- $\beta + \gamma < 1$

La Figure 18 montre les résultats de l'utilisation de triangles avec un triangle simple au centre de la scène. Alors, trouver les intersections d'un triangle sur la scène n'est pas très lourd, cependant, les maillages complexes développés en utilisant des triangles peuvent être beaucoup plus lents à calculer. Effectivement, une géométrie complexe peut être formée par des milliers de triangles, et tester les intersections d'un par un n'est pas très efficace. Pour éviter cela, nous utilisons les concepts de boîte englobante et de Bounding Volume Hierarchy.

Une boîte englobante consiste à générer un volume régulier (une boîte formée par des plans parallèles aux axes de coordonnées) pour enfermer le volume d'un maillage. Alors, à la place de chercher directement des intersections avec des triangles, nous cherchons tout d'abord des intersections avec la boîte englobante, et si l'intersection existe nous cherchons le triangle le plus proche un par un. D'un autre côté, le concept de boîte englobante peut être amélioré pour avoir un code encore plus efficace. C'est précisément cela le Bounding Volume Hierarchy : une hiérarchie de boîtes englobantes. Alors, nous avons une première boîte globale qui contient deux boîtes plus petites à l'intérieur. Chacune de ces boîtes a deux boîtes à l'intérieur et ainsi de suite. De cette manière, nous ne testons que des boîtes englobantes (faciles à tester) pour vérifier les intersections jusqu'au moment où il y a une petite quantité des triangles à tester.

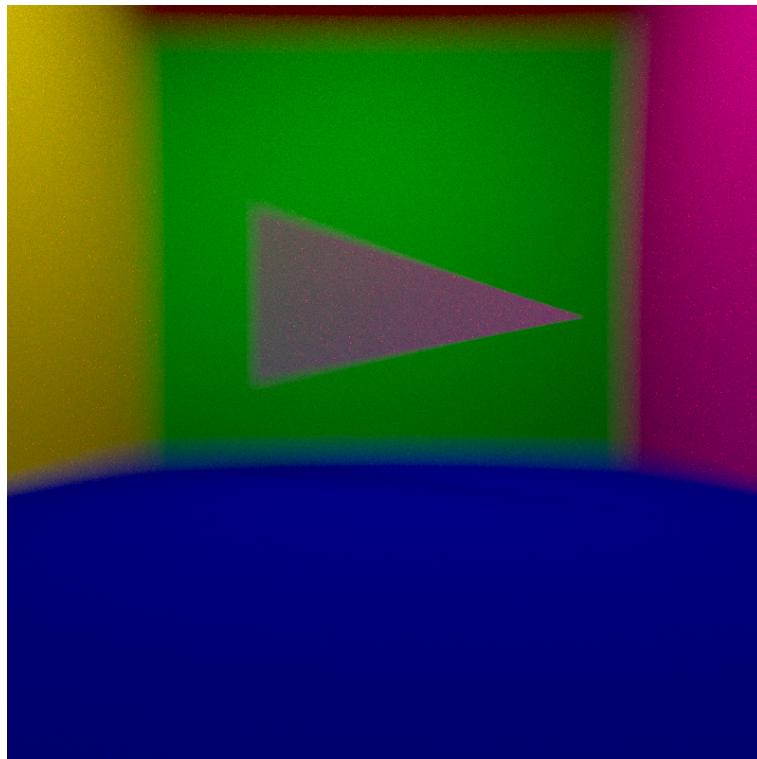


FIGURE 18 – Résultat de la gestion des intersections avec des triangles

La Figure 19 montre le résultat de la gestion des géométries. En effet, l'utilisation des boîtes englobantes n'est pas remarquable dans l'image, mais le code est disponible sur Github : [cegarciav/RayTracing](#).



FIGURE 19 – Résultat de la gestion des géométries

Nous pouvons voir dans la Figure 19 que certaines parties de la géométrie ont des ombres un peu bizarres. Les jambes de la fille sont une des parties la plus remarquable, car nous pouvons voir une ligne verticale au milieu. Pour éviter cela, nous avons utilisé le Lissage de Phong. Alors, à la place de calculer le vrai vecteur normal de chaque triangle, nous utilisons les "vecteurs normaux" à chaque sommet des triangles d'une géométrie qui sont fournis par le créateur de cette géométrie. Alors, étant donné un triangle ABC, le vecteur normal final à utiliser dans le point  $\mathbf{P}$  est donné par la formule :

$$N(P) = \alpha N_A + \beta N_B + \gamma N_C$$

où  $N_x$  est le "vecteur normal" du sommet  $x$  et  $\alpha$ ,  $\beta$  et  $\gamma$  sont les coordonnées barycentriques du triangle. La Figure 20 montre exactement la même scène que la Figure 19, mais cette fois en utilisant le lissage de Phong. La différence est facilement visible.



FIGURE 20 – Résultat de la gestion des géométries avec le lissage de Phong

Si bien les trois dernières Figures montrent des bonnes résultats, il est possible de voir qu'elles ont certaines points rouges (ou pareil) qui n'ont pas du sens. Nous n'avons pas arrivé à trouver une explication à cette erreur assez bizarre, mais il semblait d'être partie du code de l'éclairage directe. Nous avons de toute façon pensé à une manière de supprimer ces points : un filtre médiane. Le filtre médiane consiste à définir la valeur d'un pixel en utilisant son contexte. L'algorithme est décrit dans la Figure 21 : pour chaque pixel de l'image originale, nous prenons une fenêtre de valeurs (dans la Figure de 3x3 pixels). Alors, la nouvelle valeur du pixel sera la médiane des valeurs dans cette fenêtre, donc la médiane de ces 9 valeurs.

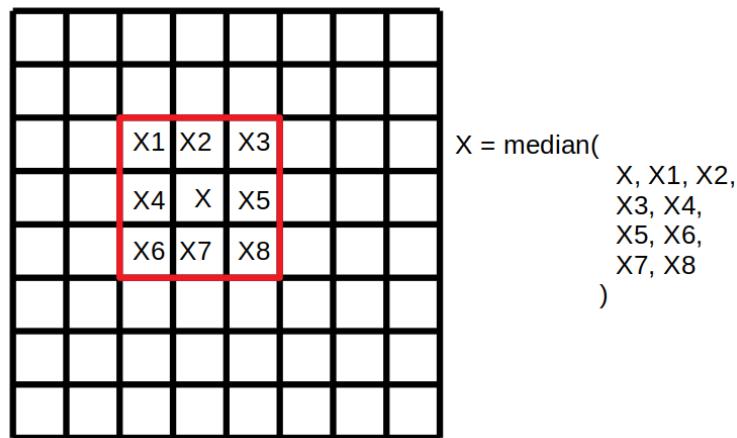
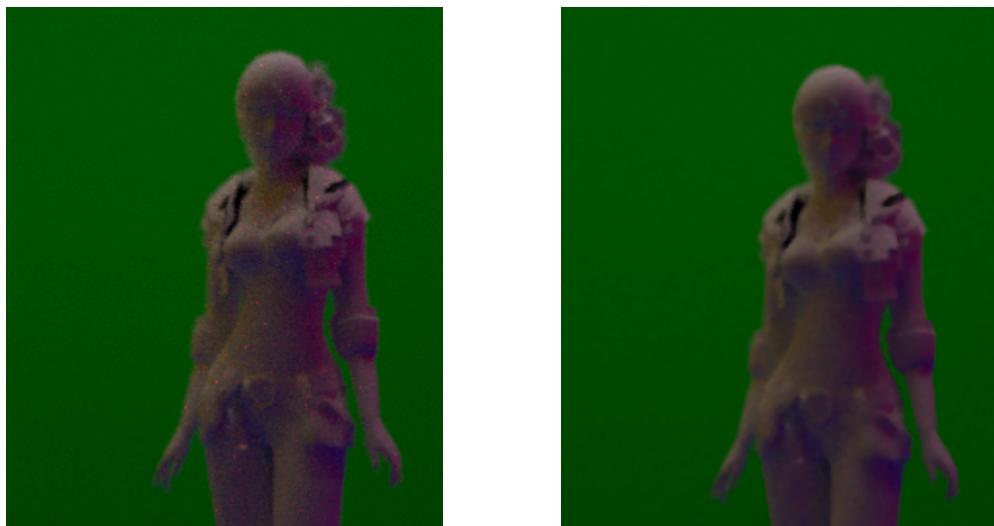


FIGURE 21 – Algorithme du filtre médiane

Alors, les Figures 22 et 23 rendent compte de l'élimination du bruit dû à l'utilisation de ce filtre médiane. En effet, il y a une légère perte de qualité, mais les résultats sont beaucoup mieux que sans le filtre.



(a) Segment d'une image sans le filtre médiane

(b) Même segment après l'utilisation du filtre médiane

FIGURE 22 – Utilisation du filtre médiane sur la fille

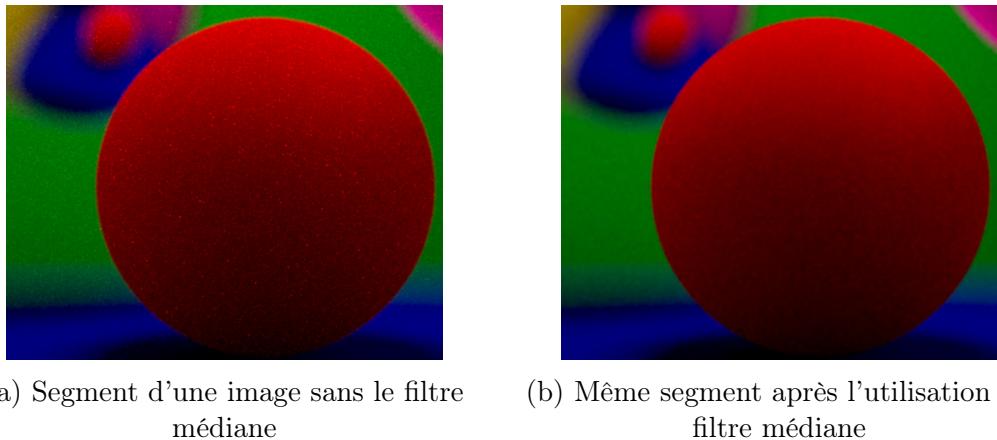


FIGURE 23 – Utilisation du filtre médiane sur des sphères

## 8 Textures, BRDF de Phong et réflexion Fresnel

Nous avons vu dans la section précédente la gestion des géométries, particulièrement une fille. Cependant, nous n'avons utilisé qu'un matériaux standard, donc la surface de cette fille était simplement une surface diffuse. Alors, ce type de géométrie que nous pouvons trouver dans l'Internet peut être accompagné par des images contenant des textures plus adéquates. La Figure 24 montre le résultat de l'utilisation des fichiers de texture sur la fille montrée dans antérieurement. Il est visible que le mapping des points des textures sur la géométrie est assez bon.



FIGURE 24 – Résultats de la gestion des textures

Comme nous avons vu également dans la sections précédente, le filtre médiane peut réduire un peu la qualité des images. Si bien nous avons trouvé que dans le cas des matériaux standard cette perte de qualité est négligeable par rapport aux améliorations obtenues, cela n'est nécessairement pas le cas des géométries avec des textures. Compte tenu que les textures ont des détails plus petits, comme les yeux de la fille, le filtre médiane peut impliquer une perte de qualité importante ici. Nous pouvons voir cette perte dans la Figure 25.



(a) Visage de la fille sans le filtre médiane   (b) Visage de la fille avec le filtre médiane

FIGURE 25 – Le filtre médiane génère des pertes de qualité pas négligeables avec les textures

En ce qui concerne les matériaux standard, nous avons pour le moment les matériaux diffus, ceux totalement spéculaires et les transparencies. La BRDF de Phong nous permet d'avoir des matériaux intermédiaires qui ne sont pas 100% diffus mais pas 100% spéculaires. La BRDF de Phong suit la formule suivante :

$$f(\vec{w}_i, \vec{w}_r) = \frac{\rho_d}{\pi} + \frac{\rho_s(\gamma + 2)}{2\pi} (\vec{R} \cdot \vec{w}_r)^\gamma$$

Cette formule indique qu'il y a une composante diffuse (le premier terme) et une partie spéculaire (le deuxième terme) qui dépend d'un facteur  $\gamma$  qui s'appelle l'exposant de Phong. Cet exposant va gérer l'amplitude d'un lobe créé autour du rayon de réflexion pure  $\vec{R}$  selon la schéma de la Figure 26

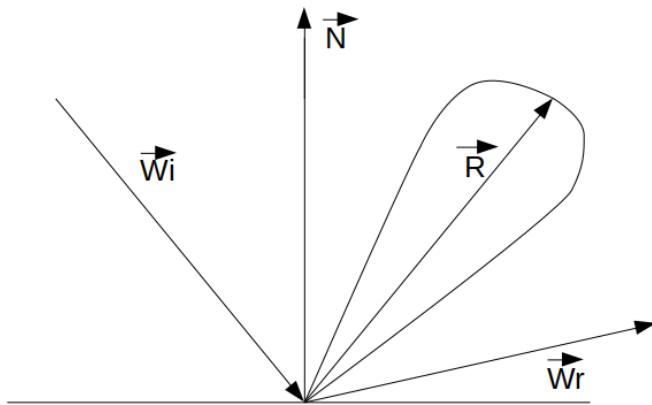


FIGURE 26 – Schéma de la BRDF de Phong

Nous pouvons voir les résultats de l'utilisation de la BRDF de Phong dans la Figure 27. En effet, dans cette scène nous trouvons trois sphères centrales avec un exposant de Phong égal pour les trois (nous avons utilisé 1000 comme valeur) mais avec des proportions différentes pour la partie spéculaire : 80%, 50% et 20% de gauche à droit. Il est clairement visible que la sphère plus à gauche est plus spéculaire que les autres deux, mais que de toute manière les trois ont au moins une faible composante spéculaire.

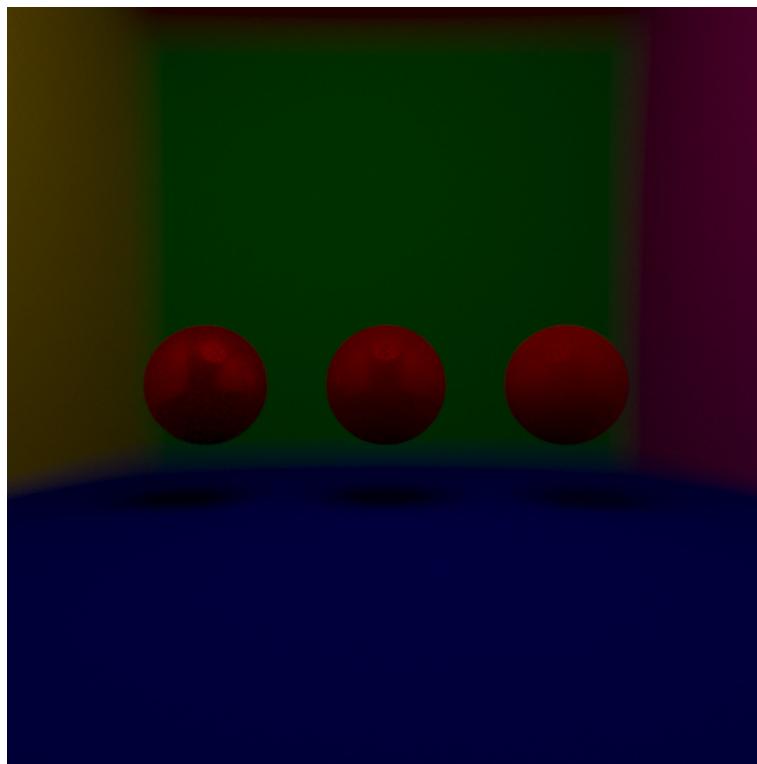


FIGURE 27 – Résultat de la gestion de la BRDF de Phong

Le dernier effet que nous avons géré est la réflexion de Fresnel. En fait, quand un rayon arrive sur une surface transparente il y a un rayon réfracté comme nous avons vu jusqu'à

maintenant, mais il y a aussi une partie réfléchie. Le calcul de la composant spéculaire dans ces situations peut être très compliqué. Cependant, nous avons utilisé l'approximation de Schlick qui donne des bons résultats et qui est très simple à calculer, parce qu'il ne dépend que des indices de réfraction des deux milieux en question. Nous pouvons voir le résultat de la gestion de la réflexion de Fresnel dans la Figure 28. En effet, nous pouvons vérifier que maintenant la sphère transparent a un certain niveau de réflexion, dont la partie la plus évidente est l'apparition d'un cercle blanc qui correspond à la source de lumière.

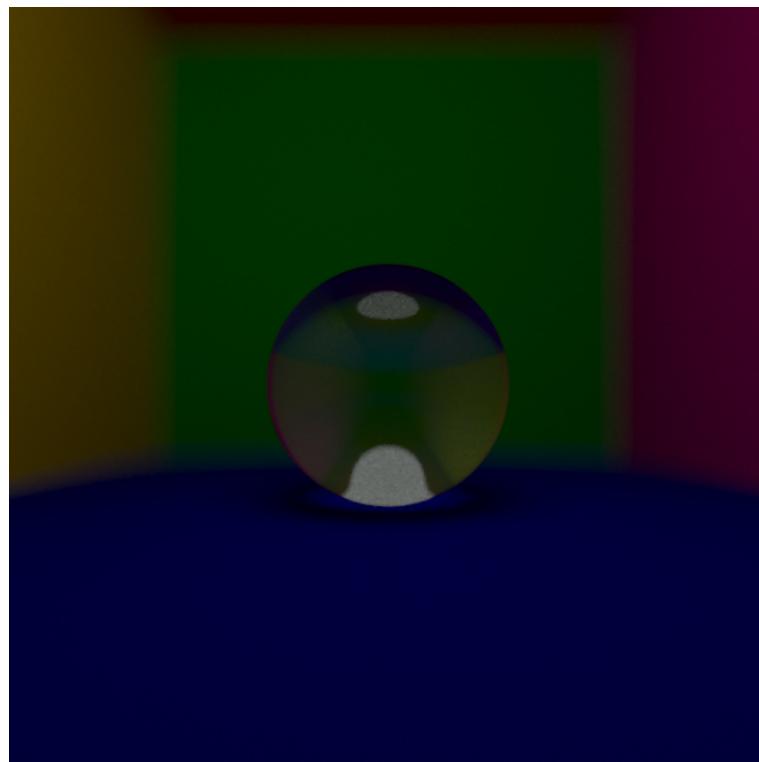


FIGURE 28 – Résultat de la gestion de la réflexion de Fresnel

## 9 Conclusion

Le Ray Tracing n'est pas une méthode pensée pour l'utilisation en temps réel. Effectivement, nous avons remarqué que les algorithmes deviennent de plus en plus lents quand nous ajoutons des nouvelles caractéristiques pour rendre les images plus réalistes. Cependant, nous pouvons constater que les résultats peuvent être assez intéressants si le temps n'est pas une limite. D'un autre côté, il faut toujours faire attention au développement des algorithmes pour essayer de les rendre aussi efficaces que possible. Par exemple, éviter de calculer des racines carrées s'il n'est pas nécessaire. La méthode de la boîte englobante est également un très bon exemple des possibilités que nous pouvons trouver pour réduire le temps de calcul des algorithmes.

Comme indiqué dans l'introduction, nous avons vu quelques contenus dans le cours qui n'ont pas été traités ici. Toutefois, le cours a été une introduction très complète à la méthode du Ray Tracing. Si bien les résultats sont assez satisfaisants, c'est vrai aussi qu'il est toujours possible d'améliorer et d'aller plus loin.