

# Rendu par lancer de rayon

---

IFT3355 : Infographie - TP #2

Jérémie DUMAS  
Baptiste DE LA ROBERTIE

17 mars 2011  
Université de Montréal

## Table des matières

<b>1</b>	<b>Lancer de rayons</b>	<b>2</b>
1.1	Modèles et caméras . . . . .	2
1.2	Performances . . . . .	3
<b>2</b>	<b>Intersections</b>	<b>4</b>
2.1	Cas d'un Cube . . . . .	4
2.2	Cas du Cylindre . . . . .	4
2.3	Cas d'un Maillage . . . . .	5
<b>3</b>	<b>Illumination locale</b>	<b>6</b>
3.1	Lumière ambiante, diffuse, spéculaire . . . . .	6
3.2	Lumière réfléchie, réfractée . . . . .	6
	<b>Conclusion</b>	<b>8</b>

# 1 Lancer de rayons

## 1.1 Modèles et caméras

Pour faire le rendu d'une image par raytracing, le principe est assez simple : pour chaque pixel de l'image à produire, on *lance un rayon* pour déterminer la couleur du point correspondant dans l'espace monde. En utilisant le principe du retour inverse de la lumière, il est en effet possible de *remonter* le chemin emprunté par la lumière et de simuler naturellement des phénomènes physiques tel que la réflexion ou la réfraction.

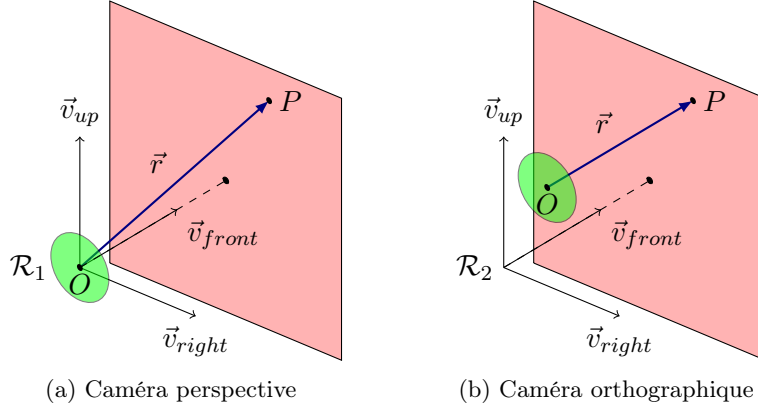


FIGURE 1: Lancer de rayon et profondeur de champ.

**Principe** On considère un point  $\tilde{P} = (x, y)$  de l'image à produire. Il correspond à un point  $P$  sur le plan de projection de la caméra (figure 1). En déprojectant  $\tilde{P}$  pour obtenir les coordonnées de  $P$ , et en calculant l'origine  $O$  du rayon à lancer, on calcule facilement la direction du lancer par  $\vec{r} \propto P - O$ .

L'origine  $O$  est calculée soit à partir du centre optique de la caméra (modèle perspective), soit à partir d'un point décalé dans le plan  $(\vec{v}_{right}, \vec{v}_{up})$  (caméra orthographique, on lance des rayons parallèles).

*Remarque.* On supposera ici que tous nos vecteurs de direction  $(\vec{r} \dots)$  sont normalisés, sauf mention contraire.

**Profondeur de champ** Afin de simuler l'effet d'une lentille et donner une impression de profondeur de champ, on ne se contente plus de lancer 1 rayon par pixel, mais plusieurs. En pratique cela signifie qu'on échantillonne la couleur donnée au point  $P$  à partir d'un nombre  $DoF$  de rayon lancés sur le point  $P$ , à partir d'une origine perturbée  $O' = O + \varepsilon$ .

La perturbation  $\varepsilon$  est calculée dans un cercle  $\mathcal{C}$  de rayon  $R_m$  dans le plan  $\mathcal{S} = (\vec{v}_{right}, \vec{v}_{up})$ . Deux façons de calculer la perturbation :

- Générer aléatoirement un rayon  $R < R_m$  et un angle  $\theta$  donnant les coordonnées polaires de  $\varepsilon$

- Générer aléatoirement les coordonnées cartésiennes  $(\hat{x}, \hat{y})$  de  $\varepsilon$  dans un carré de côté  $2R_m$  centré en  $O$ , et retirer jusqu'à ce que  $\varepsilon \in \mathcal{C}$

## 1.2 Performances

Après avoir lancé un rayon, il faut calculer les intersections avec tous les objets de la scène (voir 2), déterminer le plus proche, et calculer le shading (voir 3). Le calcul d'intersection implique de transformer le rayon dans un repère local, puis de recalculer les informations obtenues dans le repère global. Tout cela prend du temps, et en pratique on peut envisager plusieurs pistes pour améliorer les performances :

**Précalcul** En faisant l'hypothèse (raisonnable) que la scène est fixe lors de l'appel à la fonction `RayTrace`, on peut précalculer les matrices de transformation global  $global \rightarrow local$  de chaque objet de la scène. On enregistre aussi la transposée pour transformer la normale après calcul d'intersection. Cela permet de gagner presque un facteur 4 dans nos calculs.

**Résultats intermédiaires** Par soucis de clarté du code, nous avons choisi de stocker les résultats intermédiaires dans des variables temporaires. Une piste possible pour accélérer les calculs est donc de ne pas utiliser de variables intermédiaires. Cela dit l'impact réel est difficile à mesurer, même si une analyse avec `valgrind` pourrait révéler qu'il y a moins de défaut de cache L2.

**Profiling** Une analyse rapide avec `gprof` montre clairement que le programme passe le plus clair de son temps à calculer du shading (`Shader::Shade`) et des intersections (`RayTracer::FindNearestIntersection` et `Sphere::Intersect`). Par exemple sur la scène `sphere.scn` avec 50 niveaux de réflexion-réfraction, la commande :

```
make clean release LFLAGS=-pg
time ./bin/release/tp2 data/scenes/sphere.scn -i -r=50
gprof ./bin/release/tp2 gmon.out
```

Donne le résultat :

%	cumulative	self	
time	seconds	seconds	name
25.35	1.09	1.09	Shader::Shade(Node const*, Ray const&, Inter
20.23	1.96	0.87	RayTracer::FindNearestIntersection(Ray const
14.65	2.59	0.63	Sphere::Intersect(Ray const&, Intersection&)
7.44	2.91	0.32	Matrix4::operator^(Vector3 const&) const

## 2 Intersections

Dans cette partie, on considère un rayon  $\Gamma$  d'origine  $p$  et de direction  $d$ . On dira qu'un point  $m \in \mathbb{R}^3$  appartient au rayon si et seulement si  $\exists t \geq 0, m = p + td$ .

On se donne maintenant une primitive quelconque  $\Omega$ , et on cherche l'intersection entre le rayon et la primitive — c'est à dire le plus petit  $t$  (s'il existe) tel que  $p + td \in \Omega$ . Pour plus de facilités, on se placera dans le repère local de la primitive  $\Omega$ .

Chercher l'intersection nous demande aussi de renvoyer la normale à  $\Omega$  en le point d'intersection  $\tilde{m}$  calculé. Notons que si  $A$  est une matrice  $3 \times 3$  permettant de passer du repère local au repère global (sans la translation), et  $n$  la normale exprimée dans le repère local, alors la normale  $\tilde{n}$  dans le repère global se calcule par  $\tilde{n} \propto (A^{-1})^T n$ .

### 2.1 Cas d'un Cube

Les équations pour l'intersection d'un rayon avec un cube sont vraiment simple, puisqu'il s'agit essentiellement de tester 6 intersections possibles, 1 avec chacune des faces. De plus les faces ont des équations simples du genre  $z = c$ , où  $c$  est une constante. Il suffit donc de remplacer dans la définition du rayon pour déduire le paramètre  $t$  correspondant à l'intersection.

Il faut toutefois faire attention à renvoyer les normales dans le bon sens, mais surtout à tester l'appartenance du point  $m$  à une face à  $\pm\epsilon$  près, pour éviter toute instabilité numérique (par exemple quand le rayon arrive sur une arête du cube, car il risquerait sinon de passer au travers).

### 2.2 Cas du Cylindre

On s'intéresse ici à un cylindre de révolution autour de l'axe  $Oz$  du repère. Le cylindre a un rayon  $r$  et une hauteur  $h$  donnés. On dira qu'un point  $m = (x, y, z) \in \mathbb{R}^3$  appartient au cylindre s'il vérifie :

$$x^2 + y^2 = r^2, |z| \leq \frac{h}{2} \quad (1)$$

Pour déterminer la paramètre  $t$  de l'intersection (si elle existe) du rayon  $\Gamma$  et du cylindre  $\mathcal{C}$ , il suffit de projeter  $\Gamma$  dans le plan d'équation  $z = 0$ . Il ne reste alors plus qu'à calculer l'intersection entre un rayon du plan  $z = 0$  et un cercle.

En substituant dans (1) la définition du rayon projeté  $(\hat{p}, \hat{d})$ , on obtient une équation du second degré en  $t$  que l'on peut écrire :

$$\|\hat{d}\|^2 t^2 + 2(\hat{p} \cdot \hat{d})t + \|\hat{p}\|^2 - r^2 = 0$$

Les solutions sont de la forme  $t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  avec  $a > 0$ . On vérifie alors, pour les deux solutions obtenues  $t_1 \leq t_2$ , si le point d'intersection  $m = p + td$  est bien dans le cylindre (condition  $|m_z| \leq \frac{h}{2}$ ).

On note par analogie avec la sphère que si la solution  $t_2$  est retenue, cela signifie que le rayon tape à l'intérieur du cylindre (on inverse donc la direction de la normale).

## 2.3 Cas d'un Maillage

Déterminer l'intersection d'un rayon avec un maillage de faces triangulaires nécessite de calculer pour chaque triangle du maillage l'éventuelle intersection avec le rayon. Comme il peut y avoir des intersections avec plusieurs faces du maillages, il est important de renvoyer seulement la plus proche de la source du rayon.

Soit maintenant un triangle  $a, b, c \in \mathbb{R}^3$  de normale  $n$ . On commence par vérifier si le rayon n'est pas rasant (condition  $d \cdot n \neq 0$ ). Dans ce cas, après avoir déterminé le paramètre  $t$  de l'intersection  $m$  du rayon  $\Gamma$  avec le plan défini par  $(a, b, c)$ , il nous reste à vérifier si ce point est bien dans le triangle ou non.

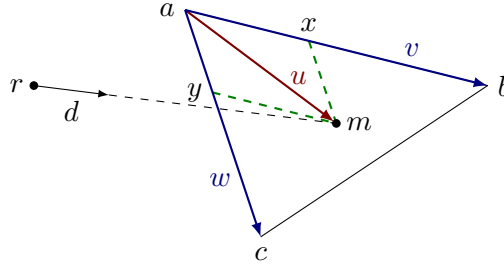


FIGURE 2: Intersection d'un rayon avec un triangle.

Pour cela on utilise les coordonnées barycentriques. Voir la figure 2 : on cherche à exprimer le vecteur  $u = m - a$  du point d'intersection en fonction des deux autres vecteurs  $v = b - a$  et  $w = c - a$  qui délimitent le triangle. On cherche donc  $(x, y) \in \mathbb{R}^2$  tels que  $u = xv + yw$ . Si  $x, y \geq 0$  et  $x + y \leq 1$ , alors  $m$  est dans le triangle, sinon il est à l'extérieur (par exemple pour  $x + y = 1$ , le point se trouve sur la droite  $(b, c)$ ).

En faisant le produit scalaire par  $v$  et  $w$ , on obtient le système suivant :

$$\begin{aligned} u \cdot v &= x\|v\|^2 + y(v \cdot w) \\ u \cdot w &= x(v \cdot w) + y\|w\|^2 \end{aligned}$$

La solution du système est donnée par les formules :

$$x = \frac{\begin{vmatrix} u \cdot v & (v \cdot w) \\ u \cdot w & \|w\|^2 \end{vmatrix}}{\begin{vmatrix} \|v\|^2 & (v \cdot w) \\ (v \cdot w) & \|w\|^2 \end{vmatrix}} \quad y = \frac{\begin{vmatrix} \|v\|^2 & u \cdot v \\ (v \cdot w) & u \cdot w \end{vmatrix}}{\begin{vmatrix} \|v\|^2 & (v \cdot w) \\ (v \cdot w) & \|w\|^2 \end{vmatrix}}$$

**Interpolation des normales** Afin de donner un rendu plus *smooth* à nos objets (sphère définie par un maillage triangulaire notamment), on utilise le modèle de Phong pour le shading. Cela signifie que la normale renvoyée pour le point  $m$  n'est pas exactement la normale  $n$  du triangle correspondant (flat shading), mais une normale interpolée entre les trois normales  $n_a, n_b, n_c$  de chaque sommet du triangle.

En réutilisant les coordonnées barycentriques  $x, y$ , on renvoie donc une normale interpolée  $\hat{n} = (1 - x - y)n_a + xn_b + yn_c$ .

### 3 Illumination locale

#### 3.1 Lumière ambiante, diffuse, spéculaire

La modèle de d'illumination locale utilisé comprend 5 parties : ambient, diffus, spéculaire, réflexion et réfraction. Le terme ambient correspond à une approximation de l'illumination globale de la scène, que l'on rend constante et identique en chaque point (approximation plausible dans le cas d'un objet convexe par exemple).

La figure 3 montre la configuration obtenue pour un rayon incident  $\vec{r}$  (direction d'où regarde l'œil) sur un objet quelconque, avec les contributions diffuses et spéculaires liées à une source de lumière  $s_i$ . On détermine d'abord si la source de lumière contribue au point  $P$  (ombrage). Pour cela il suffit de lancer un rayon depuis  $P$  dans la direction  $\vec{l}_i$  : si le premier objet rencontré se trouve *avant*  $s_i$ , alors  $P$  est dans l'ombre.

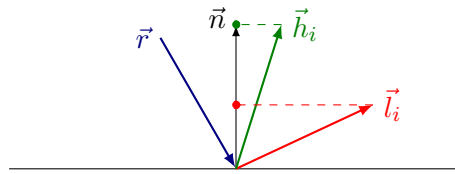


FIGURE 3: Illumination diffuse et spéculaire.

La partie diffuse de l'illumination est une fonction de  $\vec{n} \cdot \vec{l}_i$ , et la partie spéculaire est une fonction de  $\vec{n} \cdot \vec{h}_i$ , où  $\vec{h}_i \propto \vec{l}_i - \vec{r}$ . On distingue aussi pour un objet la réflexion plastique (selon la couleur de la lumière) et la réflexion métallique (selon la couleur de sa surface).

#### 3.2 Lumière réfléchi, réfractée

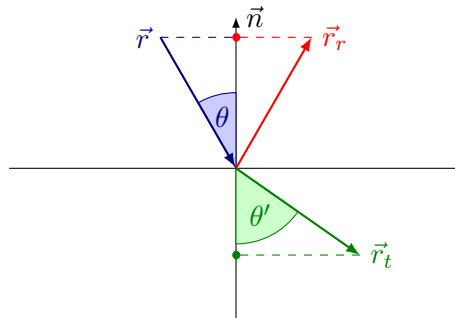


FIGURE 4: Rayons réfléchi et réfracté.

La réflexion et la réfraction sont aussi deux autres contributions que l'on prend en compte, mais seulement si le nombre de récursion est inférieur à une certaine constante  $k_{max}$  (pour éviter de boucler entre autres). La réflexion est assez simple à calculer, le

rayon réfléchi s'exprimant par  $\vec{r}_r = \vec{r} - 2(\vec{r} \cdot \vec{n})\vec{n}$  (voir figure 4). Il reste juste à lancer un rayon d'origine  $P$  et de direction  $\vec{r}_r$  pour déterminer la contribution de la réflexion.

Pour ce qui est de la réfraction, c'est un peu plus compliqué. Avec les notations de la figure 4, les lois de Snell-Descartes nous indiquent que l'on doit avoir :

$$n_1 \sin(\theta) = n_2 \sin(\theta') \quad (2)$$

Notons alors  $n = \frac{n_1}{n_2}$ , et décomposons le rayon transmis sous la forme  $\vec{r}_t = \alpha\vec{r} + \beta\vec{n}$ . En faisant le produit scalaire par  $\vec{r}$  et  $\vec{n}$ , on obtient que :

$$\vec{r}_t \cdot \vec{r} = \alpha + \beta(\vec{n} \cdot \vec{r}) = \alpha - \beta \cos(\theta) = \cos(\theta - \theta') \quad (3)$$

$$\vec{r}_t \cdot \vec{n} = \alpha(\vec{r} \cdot \vec{n}) + \beta = -\alpha \cos(\theta) + \beta = -\cos(\theta') \quad (4)$$

En faisant  $\cos(\theta)(4) + (3)$ , on obtient :

$$\begin{aligned} \alpha(1 - \cos(\theta)^2) &= -\cos(\theta') \cos(\theta) + \cos(\theta - \theta') \\ \alpha \sin(\theta)^2 &= -\cos(\theta') \cos(\theta) + \cos(\theta) \cos(\theta') + \sin(\theta) \sin(\theta') \\ \alpha \sin(\theta)^2 &= n \sin(\theta)^2 \\ \alpha &= n \end{aligned}$$

De là on déduit la valeur de  $\beta$  :

$$\begin{aligned} \beta &= \alpha \cos(\theta) - \cos(\theta') = n \cos(\theta) - \sqrt{1 - \sin(\theta')^2} \\ \beta &= n \cos(\theta) - \sqrt{1 - n^2 \sin(\theta)^2} \end{aligned}$$

Au final, le vecteur réfracté est donc :

$$\boxed{\vec{r}_t = n\vec{r} + (n \cos(\theta) - \sqrt{1 - n^2 \sin(\theta)^2})\vec{n}} \quad (5)$$

*Remarque.* Notons que le terme sous la racine peut être négatif. Cela se produit lorsqu'il y a réfraction totale, dans le cas où le rapport  $n > 1$  (attention ici  $n$  ne désigne pas l'indice du deuxième milieu). En pratique on teste si la réfraction est totale, auquel cas on réajoute la contribution du rayon réfléchi décrit plus haut.

**Limitations** Le système du lancer de rayon permet de rendre compte simplement de ces deux phénomènes physiques que sont la réflexion et la réfraction. Cependant si on veut implanter d'autres phénomènes plus compliqués, comme les caustiques (concentration de rayons lumineux en une petite région de l'espace), cela requiert une part de réflexion supplémentaire (lancer de photon, etc.).

## Conclusion

Finalement à travers nos démarches, nous avons pu constater que le raytracing est une méthode conceptuellement simple à mettre en oeuvre. Cependant il est facile de se tromper (erreur de signe, normale transformée de façon incorrecte, calcul d'intersection erroné, etc.), donc une implémentation correcte requiert une certaine vigilance.

De plus, on soulignera l'aspect performances, qui peut être amélioré de multiples façons : calcul sur GPU, utilisation de volumes englobants pour des scènes complexes, etc. Ce n'est pas l'aspect principal de ce devoir, mais cela reste néanmoins le soucis de tout moteur de rendu par lancer de rayon, justement délaissé à ses débuts pour des raisons computationnelles.

## Références

Aucune.