

Projet : Accélération d'un Lancer de rayons basique

Le lancer de rayons est une technique utilisée en synthèse d'images pour afficher des scènes 3D complexes en calculant l'image résultant de ce que percevrait un "œil" placé dans la scène. C'est avec cette technique que sont calculées des images qui intègrent l'ombrage et les ombres portées, ainsi que la transparence et la réflexion, grâce à l'envoi dans la scène de rayons réfléchis, réfractés, et en direction des sources lumineuses.

Un modèle d'ombrage et un modèle d'éclairage sont également nécessaires pour produire des images réalistes.

La base de la technique du Lancer de rayons consiste à lancer des rayons, appelés rayons primaires, depuis l'œil vers la scène (inverses des rayons lumineux), pour déterminer le point de la surface rencontrée par ce rayon et également sa couleur, qui sera celle affichée au pixel correspondant au rayon.

Le Lancer de rayons est une technique qui a révolutionné la synthèse d'images en apportant les effets d'ombres portées, de transparence avec réfraction et de spécularité (surfaces réfléchissantes), impossibles à obtenir avec les techniques plus anciennes. Mais elle souffre d'un handicap, elle est très (très) lente Les premiers articles traitant du Lancer de rayons indiquaient un temps de calcul *estimé* de plus d'un mois, pour une scène comprenant une centaine de sphères réfléchissantes et transparentes !

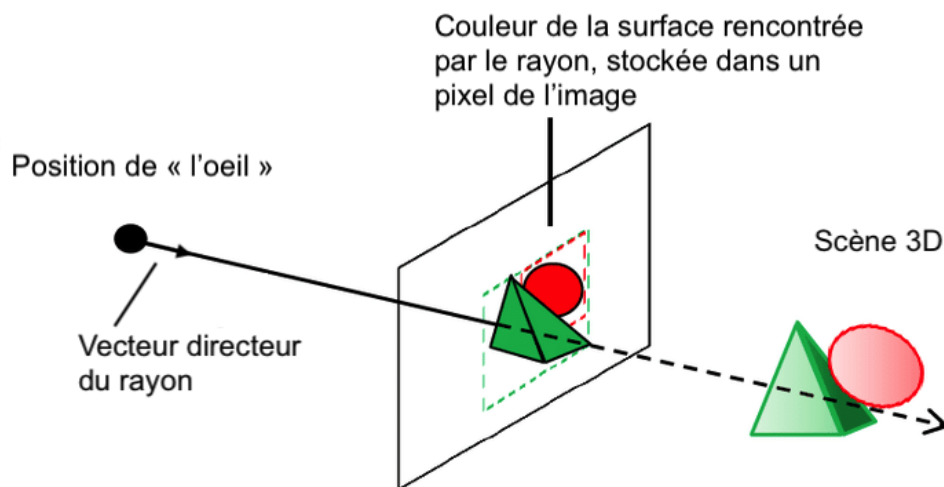


Illustration 1: Principe du Lancer de rayons (limité aux rayons primaires)

Sans chercher à comprendre et à coder l'intégralité de la technique du Lancer de rayons, le projet va se concentrer sur certains aspects relatifs à l'accélération du Lancer de rayons d'une part, et sur l'amélioration du rendu de l'image finale d'autre part.

La scène est constituée d'objets, au format Brep (Boundary representation), dont la surface est modélisée à l'aide de facettes à 3 ou 4 sommets (on se limite à ces deux cas). Certains objets peuvent être modélisés par un très grand nombre de facettes.

1. Les techniques d'accélération cherchent à éviter les calculs inutiles en évitant de calculer l'intersection rayon-objet (et donc l'intersection entre le rayon et toutes les facettes de l'objet) dans le cas où on peut déterminer au préalable que le rayon n'intersectera pas l'objet. Pour cela, on peut utiliser des volumes englobants, nécessitant peu de calculs, pour

déterminer s'ils sont intersectés par le rayon.

Si oui, alors il faut calculer l'intersection entre le rayon et les facettes de l'objet, sinon le rayon n'intersecte pas l'objet.

Proposez plusieurs types de volumes englobants. Vous devrez proposer plusieurs types de volumes englobants, et évaluer quel est le type de volume englobant à privilégier en fonction des caractéristiques des objets composant la scène, en proposant des critères de comparaison permettant un choix.

Les volumes englobants les plus courants sont la boîte (parallèle aux axes de la scène) et la sphère.

Existe t'il d'autres techniques d'accélération du lancer de rayons ?

Comparez ces différentes techniques avec la boîte et la sphère englobante, en indiquant pour chacune l'intérêt de cette technique, ses avantages et ses inconvénients.

Pour le projet, vous devrez **coder** :

- 1.1. La fonction **boiteEnglobante** qui calcule la boîte parallèle aux axes de la scène, englobant un objet passé en paramètre.
- 1.2. La fonction **sphereEnglobante** qui calcule la sphère, englobant un objet passé en paramètre.
- 1.3. La fonction **intersectionRayonSphere** qui calcule l'intersection entre le rayon (donné par son origine O et son vecteur directeur D), et la sphère donnée par son centre et son rayon (sphere.C et sphere.r).
- 1.4. Et vous devrez **prendre en compte les volumes englobants** dans l'algorithme du Lancer de rayons

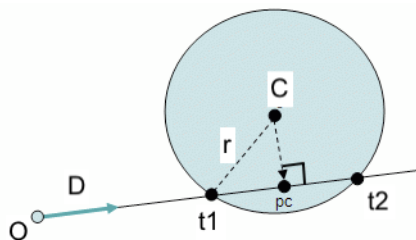


Illustration 2: Intersection rayon-sphère

L'intersection entre le rayon et la boîte englobante est identique à celle qui calcule l'intersection entre un rayon et un objet et ne sera pas à fournir.

2. Vous devrez également étudier une technique d'amélioration de la qualité de l'image produite afin d'éviter l'effet de crénelage (aliasing ou aliassage) très important que l'on constate sur les bords de l'objet dans l'image, crénelage dû à la très faible résolution de l'image finale.

Cette technique repose sur le sur-échantillonnage de l'image, qui lance n rayons pour chaque pixel, et qui détermine la couleur résultante en calculant la moyenne des n couleurs obtenues pour les n rayons lancés pour 1 seul pixel.

Vous devrez comparer les deux techniques :

- 2.1. Sur-échantillonnage régulier : les n rayons sont lancés selon une grille régulière.
- 2.2. Sur-échantillonnage aléatoire : les n rayons sont lancés aléatoirement dans la surface du pixel.

Vous évalueriez l'apport sur l'image finale de l'une ou l'autre technique en fonction du taux de sur-échantillonnage : **combien de rayons suffirait-il de lancer par pixel pour améliorer la qualité ? Au-delà de quel taux cela ne vaudrait plus la peine ?**

Pour simplifier les algorithmes, les données correspondant à une boîte englobante, appelons-la boîte, sont stockées de la même façon que les objets de la scène :

- boîte.F : 6 faces de 4 index de sommets (index dans boîte.V), [1 2 3 4 ; 5 6 7 8 ; ...] pour les 6 faces et les 24 sommets.
- boîte.N : 6 normales de 3 coordonnées (x, y, z)
- boîte.V : 24 sommets de 3 coordonnées (x, y, z), les sommets sont répétés dans V, de façon ce qu'ils soient dans l'ordre d'indexation de F, dans l'absolu il suffirait bien évidemment de stocker 8 sommets pour une boîte ...
- boîte.NA : 24 vecteurs, NA(i) est le vecteur entre les sommets V(i) et V(i+1) sauf pour NA(24) qui est le vecteur entre V(24) et V(1)
- boîte.C : 1 couleur (r, v, b), entre 0 et 255 pour chaque composante (utilisée ou non).

Vous devrez fournir un **rapport** présentant les techniques étudiées et vos analyses comparatives, apports, avantages, inconvénients, et éventuellement des évaluations quantitatives.