

RayTracer 3.0 - Objets, Lumières, Rayons et Caméras : Le RayTracer

Mise en place de la structure générale du programme : les "objets" et l'algo. général du RayTracer

Mise en place des "objets" - structuration du RayTracer

Il y a plusieurs types d'objets (au sens informatique) à mettre en place :

- ① **Les lumières** : ce sont les éléments les plus simples. En général un simple point suffit.
 - une position dans l'espace
 - une couleur (blanc par défaut)
 - éventuellement une direction et un angle pour les lumières directionnelles
- ② **Les formes** (ou objets) géométriques : n'ont aucune connaissance des autres éléments de la scène.
 - une id, référence à une forme canonique
 - 3 matrices de transformation en coordonnées homogènes (M_d, M_n, M_i)
 - une couleur et un jeu de paramètres définissant les propriétés optiques d'un matériau
 - des méthodes *spécifiques*, en particulier une fonction d'intersection Rayon - Forme Canonique.
 - autres champs utiles : structure de chaînage, d'arbre.... boîtes englobantes, paramètres divers...

Lumières et formes constituent la base de toute "scène" au sens de la synthèse d'image. La structure précédente reste valable quelle que soit l'application visée. Ce qui change ici par rapport à de la modélisation 3D "classique" comme le fait OpenGL, c'est la façon dont sont gérées les interactions Formes/Lumières. Dans le modèle de base, l'illumination d'un point d'un objet ne dépend que de la position de la lumière par rapport à ce point, pas de la position d'observation : il ne peut donc pas y avoir de reflets ni de transparence (mais il pourrait y avoir les ombres).

- ③ **Les rayons** : font le lien entre les lumières et les formes. Lorsque qu'un rayon croise une forme, il modélise, au point de contact la façon dont lumière et matière interagissent.
 - une source et une direction qui seront actualisées par la caméra ou par les réflexions / transmissions sur les formes rencontrées (récursivité).
 - un champs couleur (informations "remontées" de son parcours à travers la scène)
 - une "mémoire" de son parcours à travers la scène : "id" de la forme la plus proche, point d'impact et normale en ce point, distance à la source.... plusieurs possibilités plus ou moins "souples", plus ou moins "économiques".

- ④ **Les caméras** : objets les plus complexes et de plus haut niveau. Elles ont connaissance de tous les autres éléments de la scène.

Une caméra est un objet de la scène (presque) comme les autres, en ce sens qu'elle peut être définie par rapport à une *caméra canonique* et des matrices de transformations (en réalité, seule la matrice directe M_d est utile). Là s'arrête la ressemblance.

La caméra canonique, quant à elle, est définie par :

- un point focal (ou *oeil*) – par exemple le point $(0, 0, 1)$
- un centre (celui de l'écran) – par exemple le point $(0, 0, 0)$
- une distance focale (d_f) et une direction de visée – déduites des points précédents ($\vec{v}(0., 0., -1.)$)

- un écran – par exemple le carré de côté 2, dans le plan ($z = 0$)
- une résolution, correspondant à la taille de l'image finale (nombre de pixel sur l'écran).
- une matrice de transformation directe M_d :
 - Homothétie de rapports (r_x, r_y, d_f) où (r_x, r_y) définissent la "taille" de l'écran dans scène (on peut les laisser à 1) et d_f caractérisera la distance focale.
 - des Rotations multiples pour orienter et/ou piloter la caméra proprement. : c'est la partie un peu difficile, qui fera l'objet de quelques exercices intermédiaires
 - une Translation pour la positionner dans la scène.

La Caméra génère les *rayons primaires*, issus des pixels et de direction donnée par le vecteur oeil→pixel. Ces rayons, sans couleur au départ, se propagent récursivement dans la scène, rencontrent des objets, rebondissent de dessus et/ou les traversent, et remontent à chaque trajet une information de couleur (intensité/teinte) jusqu'au pixel.

Le Raytracer - l'algorithme complet

On considère que la scène a été initialisé : la caméra est positionnée, ainsi que les objets et les lumières.

les **rayons initiaux** : lancés par la caméra canonique - leur couleur est vide (noir).

Pour chaque pixel P_i :

- créer le rayon canonique $R_i^0 = [E, \vec{u}_i^0]$ issu de l'oeil, passant par le centre du pixel P_i
- envoyer ce rayon dans le repère global : $R_i = [A, \vec{u}]$ où $(A = M_d^{cam} \times E)$ et $(\vec{u} = M_d^{cam} \times \vec{u}_i^0)^{(1)}$.
Par défaut, $E = (0, 0, +1)$, et l'homothétie de rapport d_f , incluse dans M_d , fixera la distance focale.

👉 Ici démarre la procédure (potentiellement récursive) du RayTracer : suivi du rayon à travers la scène.

- pour chaque objet Ω de la scène, tester l'intersection avec R_i :
 - on teste en réalité l'intersection entre l'objet canonique Ω^0 et le rayon $[A^\Omega, \vec{u}^\Omega]$, projeté de R_i dans l'espace de l'objet canonique : $(A^\Omega = M_i^\Omega \times A)$ et $(\vec{u}^\Omega = M_i^\Omega \times \vec{u})^{(1)}$
 - si une intersection est détectée, on la compare à celle éventuellement déjà enregistrée par le rayon R_i . Si elle est plus proche, on enregistre cette nouvelle intersection (*grande marge de manoeuvre sur la façon de faire*), en renvoyant les informations $(I^\Omega, \vec{N}_i^\Omega)$ dans le repère global : $(I = M_d^\Omega \times I^\Omega)$ et $(\vec{N}_i = M_d^\Omega \times \vec{N}_i^\Omega)^{(1)}$
- Si il y a un objet en mémoire dans R_i , c'est le plus proche (sinon, c'est fini pour ce rayon).
On peut alors passer à la phase de "collecte" de la couleur à remonter.

Tout dépend du degré de sophistication du modèle (ces degrés seront détaillés à part) :

- **degré 0** : le rayon "remonte" simplement la couleur Col^Ω de l'objet Ω enregistré (cf. TD.3).
- **degré 1** : interaction forme/lumière (supposée blanche, uniforme, positionnée en un point L)
On forme le vecteur (normé) $\vec{w} = \vec{IL} / \|\vec{IL}\|$ et on module la couleur du rayon en fonction de l'angle d'incidence de la lumière au point I :
 - si $(\vec{N}_i \cdot \vec{w}) \leq 0$, le point I n'est pas illuminé par L. On considère alors qu'il est noir ($Col^R = 0$) ou qu'il baigne dans une lumière ambiante définie par une fraction α faible de la lumière ($Col^R = \alpha \times Col^\Omega$)
 - sinon, $Col^R = (\vec{N}_i \cdot \vec{w}) \times Col^\Omega$ (plus, éventuellement, le terme ambiant $\alpha \times Col^\Omega$)
 - on peut également introduire une atténuation, fonction de la distance $\|\vec{IL}\|$ (mais ça ne se justifie pas vraiment ici)
- **degré 2** : interaction matière/lumière
Les matériaux réagissent différemment à la lumière : certains sont très absorbants, d'autres plutôt mat ou plutôt brillants, d'autres encore sont transparents. On va donc associer à chaque objet un jeu de paramètres définissant son matériau (en plus de sa couleur). Pour l'instant il faut 2 paramètres ($ambi^\Omega$, $diff^\Omega$). Le modèle précédent est alors pondéré par ces coeff. :
 $Col^R = (\vec{N}_i \cdot \vec{w}) \times (diff^\Omega \times Col^\Omega) + \alpha \times (ambi^\Omega \times Col^\Omega)$
- **degré 3** : réflexions (objets "brillants")
Pour les objets brillant, on va lancer un nouveau rayon R_r (rayon réfléchi - récursivité) issu du point I, de direction symétrique au rayon incident R_i par rapport à la normale \vec{N}_i .

⁽¹⁾ ne pas oublier de normer le vecteur

- **degré 4 : réfractions** (objets "transparents")
 Pour les objets transparents, on va lancer un nouveau rayon R_t (rayon **transmis** - récursivité) issu du point I , de même direction générale que rayon incident R_i mais légèrement dévié (rapport de milieux).
 Ce rayon va traverser l'objet Ω et ressortir en un nouveau point (J, \vec{N}_j) .
 De là, ce rayon transmis R_t repart dans la scène avec comme source J et une direction encore légèrement déviée par le changement de milieu.
- **degré 5 : ombres** (occultations).
 La gestion des ombres intervient plus précisément au **degré 2** : le point I de Ω est "éclairé" par L si l'angle d'incidence est correct mais également si aucun autre objet ne se trouve entre I et L .
 On lance donc un rayon d'ombre R_o , depuis I , dans la direction \vec{w} et si ce rayon croise un objet, on arrête (il n'y aura pas d'illumination directe, mais il peut y avoir des réflexions et transmissions).

Le Rayon Incident R_i "accumule" ainsi les informations récoltées au niveau local (1,2,5) et remontées par les rayons réfléchis (3) et (4)

formation de l'image finale :

Chaque rayon primaire remonte ainsi la couleur du pixel qu'elle traverse. Pour former une image correcte, il est souvent nécessaire de passer par une phase de calibration (garantir que toutes les données collectées sont bien dans $[0, 1]$, avant de convertir cette grille de valeurs en vraie image (au format brut PPM, par exemple).

Pour améliorer la qualité, on peut également lancer plusieurs rayons par pixel.

