Université de Bordeaux 1 - LP DAWIN

Maths pour l'image : algèbre linéaire et géométrie

Projet de programmation : Rendu par lancer de rayon

fourer@labri.fr - Novembre 2011

1 Introduction

Le rendu par lancer de rayon ou « raytracing » est une technique de synthèse d'image qui simule le parcours inverse de la lumière.
Il permet ainsi de réaliser de façon très réaliste des effets d'optique : ombres, illumination d'objet, phénomène de réflexion et de réfraction.

Le principe est le suivant :

1. un rayon primaire est lancé depuis chaque pixel (plan de projection de la caméra) vers la scène 3D,
2. si le rayon intersecte un objet, de nouveaux rayons sont émis depuis l'objet vers chaque source lumineuse.
3. La couleur définitive du pixel à calculer est donnée par le modèle d'illumination utilisé (en général, l'énergie totale accumulée au point provenant des sources lumineuses).

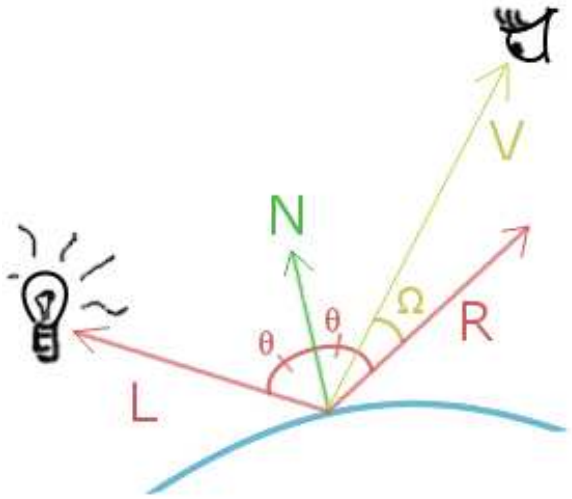
1.1 Calcul de l'intersection avec un objet

Chaque rayon r est défini par une origine O_r et une direction $\vec{d_r}$. Le rayon intersecte un objet si il existe $\lambda \in \mathbb{R}$ tel que le point $P = O_r + \lambda \vec{d_r}$ appartient à un objet. Chaque objet étant défini implicitement par son équation, on vérifie alors que P soit solution de cette équation. Dans le cas où il existe plusieurs λ solution de l'équation d'un ou de plusieurs objets, on conserve alors le λ minimal (P le plus proche de la caméra).

1.2 Modèle d'illumination de Phong

Dans le cas où le rayon secondaire (émis depuis l'objet vers chaque source de lumière) atteint directement la source lumineuse, on calcule alors la contribution de cette source pour définir la couleur du pixel en utilisant un modèle d'illumination.
Le modèle de Phong choisi ici est un modèle empirique relativement peu couteux qui permet d'effectuer un rendu crédible. Celui-ci est défini comme suit :

$$c = \underbrace{c_a \cdot k_a}_{\text{ambient}} + \sum_{i \in [1,S]} \underbrace{c_d \cdot k_{i,d} \cdot (\vec{L_i} \cdot \vec{N})}_{\text{diffus}} + \underbrace{c_s \cdot k_{i,s} \cdot (\vec{R} \cdot \vec{N})^{\alpha}}_{\text{spéculaire}} \quad (1)$$



S étant le nombre de sources lumineuses. $\vec{N}$ est la normale au point d'intersection, $\vec{L}_i$ la direction vers la lumière i, $\vec{R}$ le rayon réfléchi $\vec{V}$ la direction de l'observateur. k_a, $k_{i,d}$ et $k_{i,s}$ sont respectivement les coefficients d'illumination ambient, diffus et spéculaires et α est l'exposant spéculaire de brillance ou "shininess". Le terme ambient est fixe pour l'intégralité de la scène (illumination globale) et les coefficients $k_{i,d}$ et $k_{i,s}$ dépendent de source lumineuse et du matériau de l'objet.

2 Travail à réaliser

2.1 Résumé

Vous devez implémenter un petit moteur de rendu en language C/C++ qui prend en entrée un fichiere scène au format xml (cf. exemple fourni) et qui génère un fichier image représentant la scène décrite.
Ce projet est à effectuer en binome et à remettre au plus tard le **21 décembre à minuit (GMT+1)** avec un rapport de 5 pages maximum résumant les travaux effectués, les difficultés rencontrés ainsi que vos choix d'implémentation.

2.2 Matériel

Commencez par récupérer l'archive du projet initial à l'adresse :
`http://www.labri.fr/perso/fourer/Ens/1112/projet.tar.gz`.

Pour fonctionner, vous devez disposer d'un compilateur C++ et de la bibliothète Qt[1]. Pour compiler, utilisez successivement les commandes **qmake** (fourni par Qt) puis **make**. En fonction de votre architecture, vous devrez peut être mettre à jour le fichier **raych.pro** situé à la racine.

[1]Pour la documentation de la bibliothèque Qt, consultez : `http://doc.qt.nokia.com/`

L'exécution du projet doit se faire depuis le dossier **bin** via la commande :

```
raych -s <fichier_scene.xml> -c <fichier_camera.xml> -o <fichier_image>
```

2.3 Evaluation du projet

Dans sa version initiale, ce programme charge en mémoire une scène et une caméra puis crée un fichier image de couleur noire ($(0, 0, 0)$ dans l'espace RGB).

Vous devez :

1. réutiliser le code existant et comprendre son fonctionnement,
2. implémenter les fonctions permettant de calculer les intersections avec une sphère, un plan et un cube,
3. implémenter la BRDF[2] de Phong pour obtenir un rendu plus réaliste tenant compte des sources lumineuses,
4. modifier le format du fichier scene.xml afin de permettre d'appliquer sur les objets de la scène (lumière, primitive et caméra) les transformations suivantes : homotéthie, translation et rotation. Chaque objet devra alors posséder un identifiant (id) unique et pourra recevoir plusieurs transformations simultanément (appliquées dans l'ordre du fichier xml).

2.4 Bonus

Des points supplémentaires seront attribués pour :

– l'implémentation des calculs d'intersection pour les primitives cylindre, cone et triangle,
– prise en charge de la reflexions entre les objets et/ou de la réfraction (transparence).

Les choix et techniques utilisées devront apparaître dans votre rapport.

Références

[1] R. Malgouyres, *Algorithmes pour la synthèse d'images et l'animation 3D.* Dunod, 2eme edition ed., 2005.

[2]Bidirectional Reflectance Distribution Function