

## IFT3355 : Infographie Travail Pratique #2 : 20%

Disponible : Vendredi 22 février 13 :30  
Remise : Jeudi 21 mars 13 :30 au début de la démo  
Equipes : Deux étudiant(e)s

### 1 Description

Le lancer de rayon est une technique populaire pour le rendu d'images de haute qualité car il permet de simuler plus naturellement que d'autres techniques de rendu certains phénomènes comme la réflexion et la réfraction. Dans ce travail, vous devez implémenter le lancer de rayon récursif. En particulier, vous devez :

1. calculer les rayons partant de l'oeil pour une caméra perspective.(5 pts)
2. calculer l'intersection entre un rayon et un maillage de triangles dans son espace canonique.(15 pts)
3. calculer l'illumination locale selon le modèle suivant (25 pts) :

$$L_{p\lambda} = c_a L_{a\lambda} k_{a\lambda} S_\lambda + \sum_{i=1}^M \left[ \frac{L_{i,l\lambda}}{\pi r_i^2} \left( k_{d\lambda} S_\lambda (N \cdot L_i) + c_s k_{s\lambda} (m_\lambda S_\lambda + (1 - m_\lambda)) (N \cdot H_i)^n \right) \right] + k_{r\lambda} L_{r\lambda} + k_{t\lambda} L_{t\lambda}$$

pour  $\lambda \in \{R, G, B\}$  et où

- $L_{a\lambda}$  est la lumière ambiante de la scène
  - $S_\lambda$  est la couleur de la surface
  - $k_{a\lambda}$ ,  $k_{d\lambda}$ ,  $k_{s\lambda}$ ,  $k_{r\lambda}$  et  $k_{t\lambda}$  sont respectivement les coefficients de réflexion ambiante, diffuse, spéculaire, miroir et de transmission
  - $M$  est le nombre de lumière
  - $L_{i,l\lambda}$  est l'intensité de la  $i^e$  lumière
  - $r_i$  est la distance entre la lumière et le point illuminé
  - $N$  est la normale à la surface
  - $L_i$  est la direction de la lumière
  - $m_\lambda$  est le coefficient de réflexion métallique
  - $H_i$  est le vecteur bisecteur entre la direction de l'oeil et  $L_i$  (selon le modèle de réflexion spéculaire de Blinn)
  - $n$  est le coefficient de rugosité
  - $c_s = \frac{n+2}{2}$  est un coefficient de normalisation du terme spéculaire
  - $c_a$  est le coefficient d'occultation ambiante
  - $L_{r\lambda}$  est la lumière réfléchiée par miroitement
  - $L_{t\lambda}$  est la lumière transmise par réfraction (ou réflexion totale interne, s'il y a lieu); chaque matériau a son propre indice de réfraction
  - la profondeur maximum de récursion pour la réflexion et la réfraction est donnée par l'utilisateur.
4. déterminer l'ombrage et l'incorporer dans le modèle d'illumination locale (5 pts).
  5. calculer le facteur d'occultation ambiante  $c_a$  (15 pts). Ce facteur, généralement calculé par des algorithmes plus complexes de calcul d'illumination globale, est ici une approximation. Le principe consiste à échantillonner uniformément une sphère unitaire centrée sur le point dont on veut calculer l'*ambient occlusion* et de lancer un

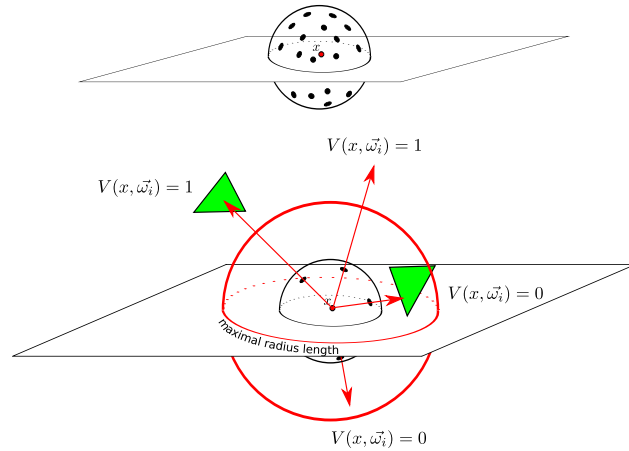


FIGURE 1 – La calcul, pour un point x, du facteur d'ambient occlusion

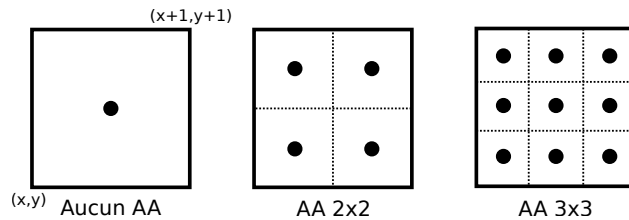


FIGURE 2 – Échantillonnage régulier d'un pixel par subdivision en sous-pixels.

rayon dont la direction est donnée par ces échantillons. Ensuite, on détermine quel est le facteur d'occultation en intersectant chacun de ces rayons avec la géométrie de la scène et ce, pour une distance maximale donnée par rapport à l'échantillon. Dans le prototype, une distance maximale de 1 a été utilisée. Le nombre d'échantillons est donné par l'utilisateur. Vous le calculerez en vous basant sur la figure 1, grâce à l'équation suivante :

$$c_a = \frac{4}{N} \sum_{i=1}^N (V(x, \vec{\omega}_i) \max(\vec{n} \cdot \vec{\omega}_i, 0))$$

6. implémenter le parcours d'une structure accélératrice, un *k-d tree*, pour le calcul d'intersection d'un maillage triangulaire (15 pts). Cette structure décompose un volume englobant le maillage en deux sous-volumes selon un plan de coupe dont la normale est un axe canonique, i.e.  $x$ ,  $y$  ou  $z$ . Cette décomposition est ensuite récursivement appliquée jusqu'à ce que le nombre de triangles par sous-volume soit sous un seuil fixé, ou jusqu'à ce qu'on atteigne un profondeur de récursion maximale. Les triangles sont stockés aux feuille de cet arbre. Étant donné un *k-d tree* déjà construit, votre travail consiste à le parcourir lors d'une requête d'intersection. Pour ce faire, vous disposez des boîtes englobantes à chacun des noeuds de l'arbre qui vous permettront de définir un critère d'arrêt lors du parcours des sous-arbres.
7. implémenter l'anti-aliasage par subdivision régulière d'un pixel (5 pts). Cette partie consiste à lancer plusieurs rayons par pixels selon un échantillonnage régulier donné par la figure 2, puis de moyennner les couleurs ainsi obtenues afin d'obtenir la couleur finale du pixel.

## 2 Support au développement

Nous vous fournissons un code de base que vous *devez* compléter. Si vous modifiez l'interface, vous devez fournir les mêmes fonctionnalités de base afin de faciliter la correction. *A priori*, vous n'avez qu'à modifier les endroits avec des commentaires pour IFT3355 dans certains fichiers. Suivez ces commentaires pour savoir quoi mettre où. Vous pouvez bien entendu ajouter de nouvelles fonctions, méthodes ou classes. L'archive du code est disponible sur le site web.

Dans la base fournie, les noeuds de la scène (classe `AbstractObject`) sont organisés sous forme d'arbre, comme pour le premier travail pratique. Chaque noeud contient une transformation globale. Un noeud de type `Geometry` contient la géométrie de l'objet et un matériau. La géométrie est essentiellement de deux types : maillage (`Mesh`) ou implicite (`Cube`, `Sphere`, `Cylinder`, `Plane`). Un maillage est simplement un ensemble de triangles, alors qu'une primitive implicite est définie par une équation quelconque. Toute géométrie est représentée dans son système de coordonnées locales, sans transformation. Vous devrez en tenir compte lors du calcul d'intersection.

Pour l'illumination, les informations nécessaires se retrouvent dans les classes `Light`, `Material` et `RayTracer`. Cette dernière contient toutes les lumières et le coefficient de lumière ambiante  $L_{a\lambda}$ , et permet de lancer des rayons lors de la réflexion, la réfraction et le calcul du coefficient d'*ambient occlusion*  $c_a$ . Chaque lumière est de type ponctuel et possède une intensité et une position dans l'espace. Les matériaux englobent le reste des coefficients du modèle d'illumination.

Le point d'entrée du lancer de rayons est la méthode `RayTracer::render`. Dans la plupart des fonctions de lancer de rayons, il y a un paramètre de type `Intersection` qui contient les informations sur l'intersection courante. Dans vos calculs, n'oubliez pas de remplir cette structure avec les bonnes informations et de les maintenir à jour lors d'un changement de repère.

Vous avez à votre disposition un prototype qui démontre le résultat final attendu.

Le code est basé sur la librairie d'interface graphique Qt. Cette librairie fournit la gestion des événements, la création de fenêtres et certains outils mathématiques. Pour plus d'informations consultez la documentation de Qt sur lesdites classes. Vous pouvez télécharger gratuitement la librairie Qt et l'éditeur *Qt Creator* sur le site web [www.qt-project.org](http://www.qt-project.org). Seule la version 4 de Qt est compatible avec le code fourni.

## 3 Fonctionnalité

Le programme dépend de ressource se trouvant dans le dossier `Data`. Il sera important de toujours exécuter votre programme ainsi que le prototype dans le dossier contenant `Data`.

Vous pouvez naviguer dans la scène à l'aide de la souris. Le bouton de gauche permet une translation selon le plan de la caméra, le bouton du milieu permet une translation dans l'axe normal au plan de la caméra et le bouton de droite permet la rotation par rapport à un point d'intérêt fixé.

Vous pouvez sélectionner la scène courante dans la liste, paramétrer le lancer de rayon et exécuter le lancer de rayon à partir de l'interface situé à droite de la fenêtre. Vous pouvez également créer vos propres scènes en héritant de la classe `Scene` et en ajoutant une instance de votre scène dans la liste de la fenêtre principale.

## 4 Délivrables et rapport du travail

Nous aurons besoin d'une version électronique de votre programme pour l'évaluer. Vous devez aussi écrire un rapport d'environ 3-4 pages qui contiendra :

1. L'énoncé du problème à résoudre.
2. La description des techniques utilisées. Donnez les formules mathématiques utilisées, et comment vous les avez employées. Nous voulons ici le fondement de vos techniques, pas les détails de la disposition du code.
3. Les problèmes reliés aux techniques implémentées, suggestions pour améliorer la solution au problème, extensions possibles.
4. Références.

La remise se fera par StudiUM.

## 5 Critères d'évaluation

La notation est sur 105 points. Tous les points au delà de 100 seront comptés comme bonus. (e.g., 103/105 → 100/100, 75/105 → 75/100)

Section	Critères	Pondération
Code	style, clarté, <i>efficacité</i> , résultats	85%
Rapport	esthétique, énoncé, descriptions	10%
Analyse	problèmes, améliorations, extensions	10%

## 6 Bibliographie

P. Shirley et S. Marschner. Fundamentals of Computer Graphics. 3ème édition, AK Peters, 2009.