



ISIR - TP5 - Matériaux et BRDFs

Maxime MARIA

2022-2023

Dans ce TP, vous allez ajouter des BRDF au moteur pour définir de nouveaux matériaux pour vos objets. Pour rappel, la BRDF, c'est la fonction f_r dans l'équation du rendu :

$$L_o = L_e + \int_{\Omega} L_i \cdot f_r \cdot \cos \theta \cdot d\omega$$

La scène utilisée pour illustrer les exercices suivants est configurée telle que :

- la caméra est située à l'origine et regarde dans la direction de l'axe z ;
- une sphère grise (GREY) et de rayon 1 est située en $(0, 0, 3)$;
- un plan rouge (RED) orienté vers y est situé en $(0, -2, 0)$;
- une lumière ponctuelle de couleur blanche et de puissance 60 est située en $(0, 0, -2)$.

Si vous lancez le rendu tel quel, vous aurez une sphère beaucoup trop lumineuse, c'est normal !

1 Matériau parfaitement diffus : modèle de Lambert

Le modèle de Lambert représente des surfaces purement diffuses, pour lesquelles la lumière est réfléchi de manière uniforme dans toutes les directions. La BRDF associée est plus que simple à évaluer puisqu'il s'agit d'une constante (cf. Équation 1). Elle est donc indépendante des directions d'observation et d'incidence.

$$f_r = \frac{k_d}{\pi} \quad (1)$$

où k_d est le coefficient de réflectance diffus, représenté ici par un triplet RGB.

1. Vous trouverez deux fichiers, nommés `lambert_material.hpp` et `lambert_brdf.hpp`, respectivement dans les dossiers `materials` et `brdfs`. La classe `LambertMaterial` possède un attribut de type `LambertBRDF`. Cet attribut est utilisé dans la méthode `shade` pour appeler la méthode `evaluate` qui calcule la BRDF telle que définie par l'équation 1.
2. Appliquez ce nouveau matériau aux deux objets de la scène et lancez votre programme. Vous devriez obtenir l'image 1 (pensez à appliquer la fonction `shade` dans l'intégrateur).



FIGURE 1 – Résultat de l'exercice 1.

2 Matériau diffus et rugueux : modèle d'Oren-Nayar

Dans la réalité, aucune surface n'est vraiment parfaitement diffuse, elles sont toutes au moins un petit peu rugueuses. Le modèle d'Oren-Nayar [ON94] permet de représenter des matériaux mats en prenant en compte cette rugosité. Il assume que la surface est entièrement composée de micro-facettes formant des « V ». La rugosité de la surface est déterminée par un unique paramètre : la pente des facettes $\sigma \in [0; +\infty)$. L'équation 2 donne la BRDF associée pour une direction d'incidence $\omega_i = (\theta_i, \phi_i)$ et une direction d'observation $\omega_o = (\theta_o, \phi_o)$. Pour un rappel sur les coordonnées sphériques : https://mathinsight.org/spherical_coordinates.

$$f_r = \frac{k_d}{\pi} \cdot (A + (B \cdot \max(0, \cos(\phi_i - \phi_o)) \cdot \sin \alpha \cdot \tan \beta) \quad (2)$$

où :

$$A = 1 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33} \quad ; \quad B = 0.45 \frac{\sigma^2}{\sigma^2 + 0.09} \quad ; \quad \alpha = \max(\theta_i, \theta_o) \quad ; \quad \beta = \min(\theta_i, \theta_o)$$

1. En vous inspirant des classes `LambertMaterial` et `LambertBRDF`, créez les classes `MatteMaterial` et `OrenNayarBRDF` pour représenter des objets mats.
2. Implémentez la fonction `evaluate` qui, à partir de la normale à la surface et des directions d'observation et d'incidence, retourne le résultat de la BRDF.
3. Testez votre code en appliquant le matériau `MatteMaterial` à la sphère, vous devriez obtenir les images de la figure 2. Remarquez que pour $\sigma = 0$, le modèle d'Oren-Nayar correspond à celui de Lambert.

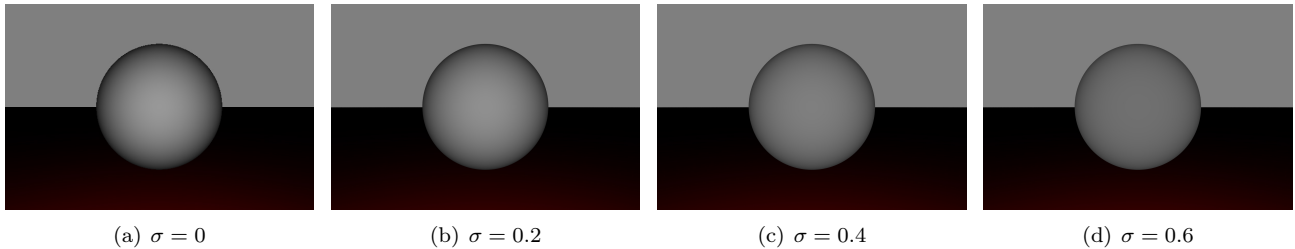


FIGURE 2 – Résultat de l'exercice 2 pour différentes valeurs de σ .

3 Matériau « plastique » : modèle de Phong

Le modèle de Phong [Pho75] propose de représenter le reflet spéculaire comme un cône aligné dans la direction miroir (ω_r) à la direction d'incidence (ω_i). Ce modèle est totalement empirique et loin d'être physiquement réaliste. Cependant, de part sa simplicité, il est encore assez utilisé, notamment dans les environnements temps-réel. L'équation 3 donne la BRDF correspondante.

$$f_r = \frac{k_d}{\pi} + \frac{k_s}{\cos \theta_i} \cdot \cos \alpha^s \quad (3)$$

où k_s est le coefficient de réflectance spéculaire (représenté ici par un triplet RGB), α est l'angle entre la direction d'observation (ω_o) et ω_r , θ_i est l'angle entre la direction d'incidence (ω_i) et la normale, et s est un coefficient permettant de contrôler la brillance du matériau (s détermine l'ouverture du cône, donc plus il est grand, plus le lobe spéculaire sera long et fin).

N.B. : Le modèle de Phong original n'est pas une BRDF à proprement parlé, il s'agit d'un modèle d'éclairage local qui ne s'intègre pas directement dans l'équation du rendu, c'est pourquoi ici nous divisons k_s par θ_i . Notez que la partie diffuse est strictement la même que dans le modèle de Lambert.

Le principal défaut de ce modèle est que, normalement, le reflet spéculaire se déforme en fonction de la direction d'incidence. Or, le modèle de Phong est totalement symétrique. Pour pallier ce problème, le modèle a été reformulé (cf. Exercice 5, optionnel).

1. Ajoutez une classe `PhongBRDF` et implémentez sa méthode `evaluate` qui n'évaluera que la partie spéculaire de la BRDF (vous pouvez utiliser `glm::reflect` pour calculer ω_r .)
2. Ajoutez une classe `PlasticMaterial` utilisant deux BRDF, `LambertBRDF` et `PhongBRDF`, calculant respectivement les parties diffuse et spéculaire de la BRDF.
3. Testez votre code et comparez vos résultats avec la figure 3 qui montre une sphère grise (70% diffuse et 30% spéculaire) pour plusieurs valeurs de s .

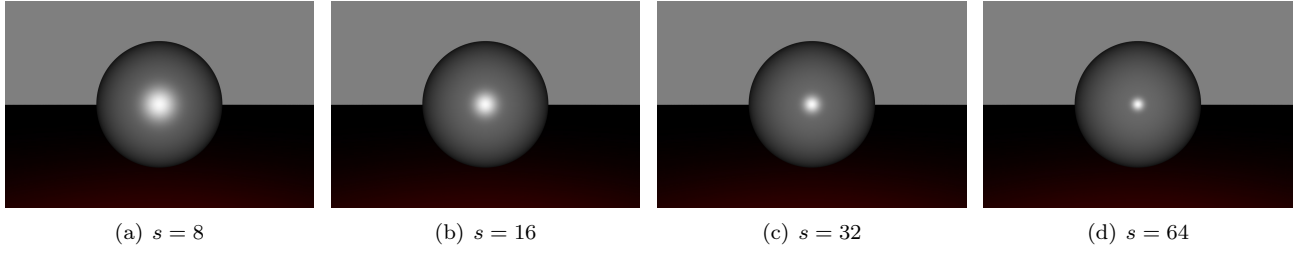


FIGURE 3 – Résultat de l'exercice 3 pour différentes valeurs de s .

4 Matériau physiquement réaliste : modèle de Cook-Torrance

Vous allez maintenant implémenter une BRDF à micro-facettes (*cf.* cours) en utilisant le modèle de Cook-Torrance [CT82]. L'équation 4 donne la partie spéculaire de la BRDF (pour la partie diffuse, nous utiliserons le modèle d'Oren-Nayar, si vous l'avez codé, sinon Lambert suffira).

$$f_{rs} = \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)} \quad (4)$$

où :

- D est la fonction de distribution des normales (*NDF*) des micro-facettes. Elle donne une approximation de la proportion de surface des micro-facettes qui sont alignées avec « demi-vecteur » $h = \frac{\omega_o + \omega_i}{\|\omega_o + \omega_i\|}$. Cette proportion dépend de la rugosité de la surface.
- G est la fonction de masquage/ombrage qui modélise le fait que la lumière soit bloquée à l'intérieur des micro-facettes.
- F est l'équation de Fresnel qui décrit la proportion de réflexion de la surface pour un angle donné. La rugosité de la surface est toujours contrôlée par σ .

Dans cet exercice, nous allons utiliser les mêmes fonctions D et F que dans le moteur de rendu Unreal Engine 4 d'Epic Games¹. Nous notons $\alpha = \sigma^2$.

Pour D , nous choisissons donc d'utiliser la distribution de Trowbridge-Reitz/GGX [TR75, WMLT07]², définie telle que :

$$D(h) = \frac{\alpha^2}{\pi((n \cdot h)^2(\alpha^2 - 1) + 1)^2} \quad (5)$$

Pour G , nous choisissons donc d'utiliser le modèle de Smith [Smi67] pour GGX, tel que :

$$G(\omega_i, \omega_o, n) = G_1(n \cdot \omega_o)G_1(n \cdot \omega_i) \quad (6)$$

$$G_1(x) = \frac{x}{x(1 - k) + k} \quad (7)$$

$$k = \frac{(\sigma + 1)^2}{8} \quad (8)$$

Enfin pour F , nous choisissons d'utiliser l'approximation de Schlick [Sch94] de l'équation de Fresnel :

$$F(\omega_o, h, F_0) = F_0 + (1 - F_0)(1 - (h \cdot \omega_o))^5 \quad (9)$$

où F_0 est la réflectance dans la direction de la normale.

1. Ajoutez les classes nécessaires à la gestion d'un matériau associé à une BRDF à micro-facettes comme décrit précédemment. Vous ajouterez un attribut `_metalness` à la classe du matériau. Celui-ci permettra de régler le côté métallique de la surface.
La réflectance finale sera calculée telle que : $(1 - \text{_metalness}) * \text{diffuse} + \text{_metalness} * \text{specular}$.
2. Testez votre code avec un matériau doré ($F_0 = (1, 0.85, 0.57)$) et une rugosité de 0.3. Vous devriez obtenir les résultats illustrés sur la figure 4.

1. https://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_notes_v2.pdf

2. Let's stop calling it « GGX » : <https://pharr.org/matt/blog/2022/05/06/trowbridge-reitz>

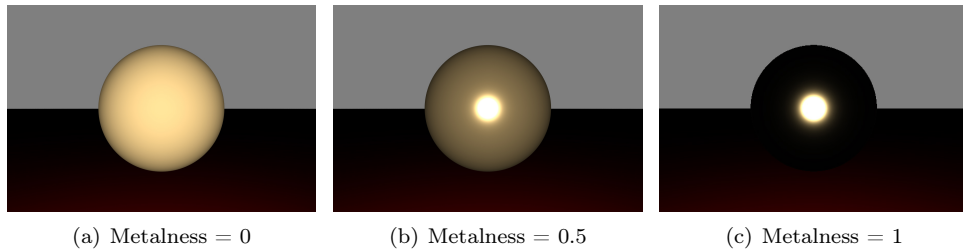


FIGURE 4 – Résultat de l'exercice 4.

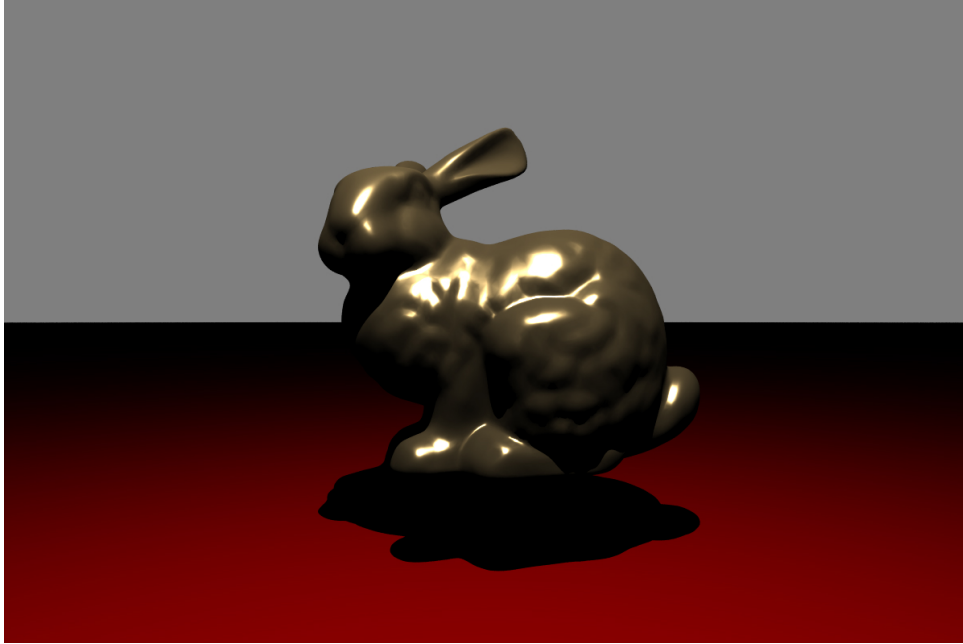


FIGURE 5 – Golden Bunny !

5 (Optionnel) Modèle de Blinn-Phong

Comme évoqué dans l'exercice 3, le modèle de Phong a pour principal défaut d'être parfaitement symétrique alors que le reflet spéculaire est censé se déformer en fonction de l'incidence. Pour pallier ce problème, le modèle de Blinn [Bli77] (souvent appelé Blinn-Phong) reformule celui de Phong en remplaçant l'angle α par l'angle formé par la normale à la surface et le « demi-vecteur » h . La figure 6 compare les deux modèles.

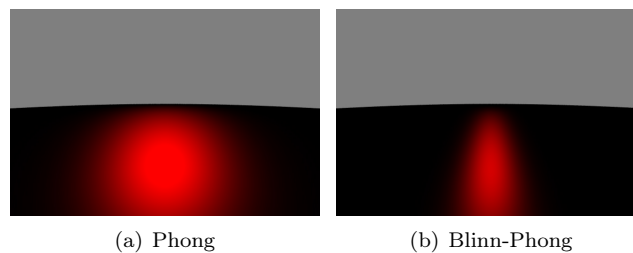


FIGURE 6 – Comparaison de la réflexion spéculaire entre le modèle de Phong et celui de Blinn-Phong.

1. Vous savez ce qu'il vous reste à faire...

Références

- [Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2) :192–198, July 1977.
- [CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1) :7–24, January 1982.
- [ON94] Michael Oren and Shree K. Nayar. Generalization of lambert’s reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’94, pages 239–246. ACM, 1994.
- [Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6) :311–317, June 1975.
- [Sch94] Christophe Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13(3) :233–246, 1994.
- [Smi67] B. Smith. Geometrical shadowing of a random rough surface. *IEEE Transactions on Antennas and Propagation*, 15(5) :668–671, 1967.
- [TR75] T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *Journal of the Optical Society of America (1917-1983)*, 65(5) :531, May 1975.
- [WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR’07, page 195–206, Goslar, DEU, 2007. Eurographics Association.