

OPENSOURCE3D et OGREMAX

Définitions



SHADING / OMBRAGE :

L'**ombrage plat** (en anglais *flat shading*) ou **ombrage de Lambert** est une technique de simulation d'éclairage dans un rendu 3D.

Il consiste simplement à affecter à chacune des faces de l'objet un éclairage proportionnel à l'angle entre la normale à la face, estimée plane, et la direction de la source lumineuse qui éclaire la scène. Cette valeur est constante sur toute la face d'où la discontinuité d'ombrage aux bords de chaque face.



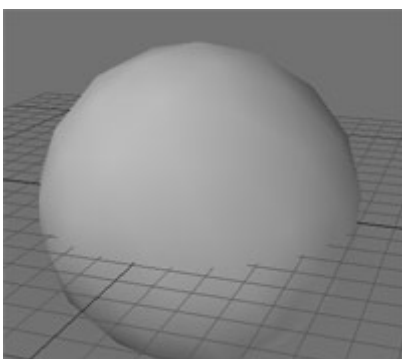
L'**ombrage de Gouraud** (*Gouraud shading* en anglais) est une technique de rendu 3D inventée par Henri Gouraud. Autrefois limitée au monde de l'image de synthèse, l'ombrage de Gouraud est aujourd'hui utilisé par toutes les cartes 3D du marché.

Cette technique marque une énorme avancée sur l'ombrage plat autrefois utilisé. Avec l'ombrage plat les facettes des objets 3D sont toujours très visibles. L'ombrage de Gouraud consiste à interpoler linéairement la luminosité entre les 3 sommets d'un polygone. Appliqué à chaque sommet, cet ombrage plus réaliste va adoucir les angles, d'autant plus qu'il est à l'origine d'un dégradé d'intensité. Comme toute technique travaillant sur l'intensité lumineuse, il est possible de combiner l'ombrage de Gouraud avec une texture en modulant la valeur des texels.

Contrairement à l'ombrage plat qui, comme son nom l'indique, produit la même intensité lumineuse pour toute une face, l'ombrage de Gouraud interpole linéairement l'intensité lumineuse pixel par pixel sur la surface dans l'espace 2D écran. La méthode la plus courante consiste à remplir le polygone ligne par ligne. L'interpolation étant linéaire et la surface considérée étant plane, la différence d'intensité d'un pixel à l'autre est constante sur une même face.

L'interpolation linéaire n'est toutefois pas toujours satisfaisante. On observe un effet appelé bandes de Mach, une illusion d'optique causée par des différences de contraste qui tendent à accentuer les faces. L'ombrage de Gouraud n'est pas non plus capable de représenter correctement la lumière spéculaire. Une source lumineuse dont la projection sur la face se trouve proche du centre ne pourra être complètement considérée par ce modèle. L'ombrage de Gouraud interpolant à partir des sommets, l'intensité au centre dépendra de ceux-ci, ce qui constitue une grossière approximation de la réalité.

Pour des raisons esthétiques, l'ombrage de Gouraud est aujourd'hui dépassé par une de ses améliorations, l'ombrage de Phong, pour un coût cependant nettement supérieur.



L'ombrage de Phong

Aujourd'hui, le terme ombrage de Phong est utilisé à la fois pour le modèle d'illumination de Phong, et l'interpolation de Phong, deux algorithmes de traitement 3D en infographie. Tous les deux furent développés par Bui Tuong Phong.

Modèle d'illumination de Phong

L'illumination de Phong est un modèle local, c'est-à-dire qui calcule l'intensité en chaque point, et qui combine trois éléments - la lumière diffuse (modèle lambertien), la lumière spéculaire et la lumière ambiante.

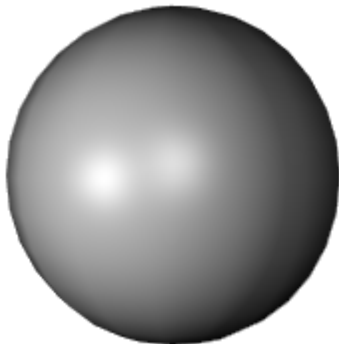
L'interpolation de Phong

Cette méthode, à mettre en parallèle avec l'interpolation de Gouraud, produit de bons résultats de rendus, souvent plus réalistes que son prédécesseur.

Le principal problème de l'ombrage de Gouraud, c'est qu'il ne calcule que les sommets (vertex) des polygones : une source lumineuse spéculaire placée au centre d'un triangle n'apparaîtra pas. Ce problème est réglé avec l'interpolation de Phong.

Soient trois sommets distincts : v_1 , v_2 et v_3 , ayant pour vecteurs unitaires normaux n_1 , n_2 et n_3 . Comme pour l'interpolation de Gouraud, celle-ci se fait de façon linéaire sur toute la surface du triangle $v_1v_2v_3$, seulement elle se fait depuis les trois vecteurs normaux des sommets, c'est-à-dire que nous interpolons en fait les vecteurs normaux au lieu des couleurs.

À la différence de l'interpolation de Gouraud cependant, le calcul ne se fait pas sur 3 points par surface, mais pour tous les points d'une surface - ou plus raisonnablement, sur plusieurs subdivisions de points. Cette méthode, bien plus lente, est parfois traitée par le matériel directement, via les shaders.



SHADERS :

Un shader (anglais, du verbe to shade : ombrager ou estomper, nuancer) est un programme utilisé en image de synthèse pour paramétrer une partie du processus de rendu réalisé par une carte graphique ou un moteur de rendu logiciel. Ils peuvent permettre de décrire l'absorption et la diffusion de la lumière, la texture à utiliser, les réflexions et réfractions, l'ombrage, le déplacement de primitives et des effets post-traitement. Par la conception même du processus de rendu, les shaders sont les candidats idéaux pour une exécution parallèle par les processeurs graphiques multiples, situés dans une carte vidéo, permettant un traitement vectorisé soulageant l'unité centrale pour obtenir un résultat plus rapidement.

Les shaders sont flexibles et efficaces. Des surfaces apparemment compliquées peuvent être rendues à partir de géométrie simple. Par exemple, un shader peut être utilisé pour générer un carrelage en céramique à partir de la simple description d'un plan.

BLEND / MELANGE :

Le blending (mélange) définit la façon de combiner une source de couleur avec la couleur de destination déjà prête à être rendue.

	Description
Zero	Chaque composante de la couleur est multipliée par (0, 0, 0, 0).
One	Chaque composante de la couleur est multipliée par (1, 1, 1, 1).
SourceColor	Chaque composante de la couleur est multipliée par la source de couleur.
InverseSourceColor	Chaque composante de la couleur est multipliée par l'inverse de la couleur source.
SourceAlpha	Chaque composante de la couleur est multipliée par la valeur alpha de la source.
InverseSourceAlpha	Chaque composante de la couleur est multipliée par l'inverse de la valeur alpha de la source.
DestAlpha	Chaque composante de la couleur est multipliée par la valeur alpha de la destination.
InverseDestAlpha	Chaque composante de la couleur est multipliée par l'inverse de la valeur alpha de la destination.
DestColor	Chaque composante de couleur est multipliée par la couleur de destination.
InverseDestColor	Chaque composante de la couleur est multipliée par l'inverse de la couleur de destination.

DEPTH / PROFONDEUR

En infographie, le Z-buffer ou tampon de profondeur est une méthode employée dans le cadre de l'affichage d'une scène 3D. Le Z-Buffer permet de gérer le problème de la visibilité qui consiste à déterminer quels éléments de la scène doivent être rendus, lesquels sont cachés par d'autres et dans quel ordre l'affichage des primitives doit se faire.

Il est principalement utilisé par l'accélération matérielle de la 3D, mais l'est aussi dans beaucoup de moteurs 3D logiciels. L'algorithme du peintre est une autre solution pour résoudre en partie le problème de visibilité.

Lorsqu'un objet est dessiné par une carte accélératrice 3D, la profondeur d'un pixel (coordonnée Z) est stockée dans un tampon (en anglais 'buffer', d'où Z-buffer). Ce tampon est en général un tableau à deux dimensions (X et Y), chaque élément étant un pixel à l'écran. Si un autre élément de la scène doit être affiché aux mêmes coordonnées (X,Y), la carte compare les deux profondeurs (Z), et n'affiche que le pixel le plus proche de la caméra. La valeur Z de ce pixel est ensuite placée dans le tampon de profondeur, remplaçant donc l'ancienne. Au final, l'image dessinée reproduit la perception de la profondeur habituelle et logique, l'objet le plus proche cachant les plus lointains.

Habituellement, la précision du Z-buffer n'est pas linéaire suivant le point, proche ou lointain. Les valeurs proches sont plus précises (de façon à afficher les objets proches plus finement) que les valeurs lointaines (moins importantes visuellement). En général c'est le comportement voulu, mais cela peut amener des artefacts visuels au fond de la scène. Une variation de la technique du Z-buffer règle ce problème, elle s'appelle le W-buffer.

Au début de l'affichage d'une nouvelle scène, le Z-buffer doit être initialisé avec une valeur prédéfinie qui représente la profondeur maximale, en général zéro.



Une scène 3d simple



Le Tampon de profondeur

DEPTH FUNCTION

Lorsque la profondeur est activée, une comparaison entre la valeur de profondeur du pixel à afficher et les valeurs déjà dans la mémoire est effectuée. Par défaut, la comparaison est de type « less_equal » c'est-à-dire que le pixel est écrit si il est plus proche (ou à la même distance) que le contenu déjà traité.

Voilà les valeurs possibles :

always_fail : Ne jamais rendre

always_pass : Toujours rendre

less : Rendu si le pixel est plus proche que le contenu existant

less_equal : Rendu si le pixel est plus proche ou à la même distance que le contenu existant

equal : Rendu si le pixel est à la même distance que le contenu existant

not_equal : Rendu si le pixel n'est pas à la même distance que le contenu existant

greater_equal : Rendu si le pixel est plus éloigné ou à la même distance que le contenu existant

greater : Rendu si le pixel est plus éloigné que le contenu existant

ALPHA REJECTION

Façon dont les pixels sont calculés lors de matériaux transparents.

Le paramètre de la fonction peut être l'une des options énumérées dans `depth_function`. La valeur de paramètre peut théoriquement être n'importe quelle valeur entre 0 et 255, mais le mieux est d'utiliser des valeurs de 0 à 128 pour la compatibilité matérielle.

LIGHTING

Définit si l'éclairage de la scène est utilisé ou non pour illuminer l'objet.

Lorsque l'éclairage est activé, les objets sont éclairés en fonction de leur « vertex normals » spéculaire et diffuse de la lumière, et globalement pour l'ambient et l'émissif.

Lorsque l'éclairage est désactivé, l'objet sera entièrement éclairé.

Ce réglage n'a pas d'effet si un vertex program est utilisé.

ITERATE

Par défaut la lumière est appliquée une fois sur un matériau, on peut grâce à la fonction `iterate`, modifier ce comportement. Soit la lumière est passée une fois de manière globale : once, soit une fois par lumière, soit avec un nombre `n` de passes.

BITMAP TYPE

Il y a 4 types d'images :

1D : C'est une texture qui fait 1 pixel de haut

2D : Fichier image classique

3D : C'est une texture qui a une hauteur et une largeur de base 2 et une profondeur. Ces textures peuvent être rendues avec des coordonnées 3d dans le cas d'une utilisation d'un shader.

Cubic : C'est une texture composée de 6 textures 2D qui sont collées dans un cube. Elles sont utilisées dans le cadre d'une réflexion par exemple.

COLOR BLEND / ALPHA BLEND

Permet le contrôle du mélange entre la texture actuelle et les textures précédentes lors d'un multitexturing.

Opérations disponibles :

`source1` : Utilise la source1 sans modification

`source2` : Utilise la source 2 sans modification

`modulate` : Multiplie la source1 par la source2

`modulate_x2` : Multiplie la source1 par la source2 puis par 2 en l'éclaircissant

`modulate_x4` : Multiplie la source1 par la source2 puis par 4 en l'éclaircissant

add : Ajoute la source1 et la source2

add_signed : Ajoute la source1 et la source2 et retire 0.5

add_smooth : Ajoute la source1 et la source2 et soustrait le produit

subtract : Soustrait la source2 à la source1

blend_diffuse_alpha : Utilise la valeur interpolée des sommets pour mesurer la source1 et ajouter la source2 redimensionnée par 1-alpha.

blend_texture_alpha : Comme blend_diffuse_alpha mais utilise l'alpha de la texture.

blend_current_alpha : Comme blend_diffuse_alpha mais utilise l'alpha courant des textures précédentes.

blend_manual : Comme blend_diffuse_alpha mais utilise un mélange alpha manuel spécifié dans « manual »

dotproduct : Produit scalaire de la source1 et de la source2

blend_diffuse_colour : Utilise l'interpolation de couleur venant des points pour redimensionner la source1 et l'ajout de la source2 redimensionnée par (1-couleur).

FILTERING

Permet de filtrer une texture lorsqu'on s'en approche ou que l'on s'éloigne.

Mode Advanced décoché :

Aucun : Aucun filtrage ou mipmapping n'est utilisé.

Bilinear : Une boîte de filtrage 2x2 est effectuée et un mipmap est pris sans qu'aucun filtre ne lui soit appliqué.

Trilinear : Une boîte de filtrage 2x2 est effectuée et deux mipmaps sont filtrés entre eux.

Anisotrope : Idem à trilinear, sauf que le filtre tient compte de la pente existante entre une face et la vue caméra.

Mode Advanced coché :

Nothing : (valide uniquement pour le mip) permet de désactiver le mipmapping.

Point : (valide uniquement pour le min et le max) désactive les filtres.

Linear : Filtre les pixels les plus proches de 2 par 2.

Anisotropic : (valide uniquement pour le min et le max) Filtre permettant de compenser la pente d'une face par rapport à la caméra.

ANISOTROPE

Permet d'augmenter la qualité de la texture quand celle-ci est éloignée, plus on s'éloigne d'un objet plus celui-ci va voir sa qualité diminuée, grâce à l'anisotropie la qualité sera meilleure. Plus le coefficient est important plus la correction sera bonne, mais aussi plus le calcul sera lourd à réaliser par la carte graphique et l'ordinateur.

CUBMAP

Beaucoup de réflexions spéculaire et de matériaux environnementaux s'appuie sur des données externes pour le calcul de leur apparence.

Ces données prendront la forme d'une *cubemap*, une texture qui représente un 360 degrés de la zone environnante. L'utilisation de la cubemap comme un point de prélèvement permet de mieux refléter l'environnement.

Le plus souvent une cubemap est réalisée à l'aide de 6 fichiers images :

Vue de face

Vue de gauche

Vue de droite

Vue de dessus

Vue de dessous

Vue de derrière

OUTILS

Un générateur de cubemap est disponible à l'adresse :

<http://developer.amd.com/gpu/cubemapgen/pages/default.aspx>

Pour gérer les fichiers dds et les normal maps, nvidia propose un kit pour photoshop :

http://developer.nvidia.com/object/photoshop_dds_plugins.html