

1 Illumination

1.1 Pipeline Graphique

L'illumination est une étape clé dans le pipeline graphique 3D. Elle intervient après les transformations de modélisation et avant la transformation d'affichage.

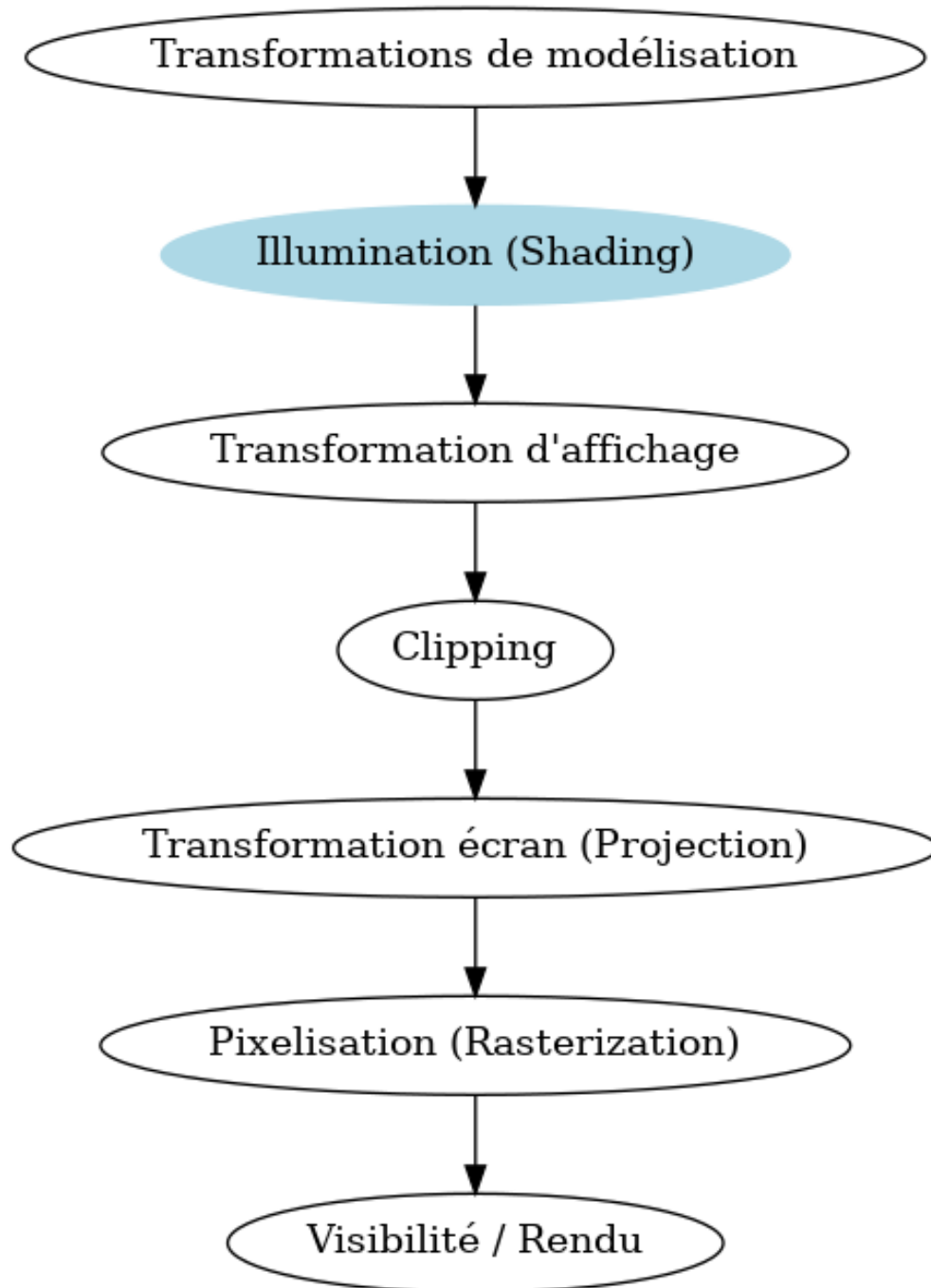


Figure 1: Place de l'Illumination dans le Pipeline Graphique.

1.2 Introduction à l'Illumination

- Les primitives (points, lignes, polygones) sont éclairées selon leur matériau, le type de surface et les sources de lumière environnantes.
- Les modèles d'illumination sont généralement locaux, ce qui signifie que le calcul de l'éclairage pour une primitive ne prend pas en compte les ombres portées par d'autres objets. Le calcul est effectué par primitive en utilisant des modèles comme diffus, ambiant, Gouraud, Phong, etc.

L'illumination comprend deux aspects principaux :

- **Ombrage (éclairage & shading):** Détermine comment la lumière interagit avec la surface de l'objet pour définir sa couleur et sa luminosité.
- **Texture (plaquage, mappage):** Applique des détails de surface (images, motifs) sur les objets pour augmenter le réalisme.

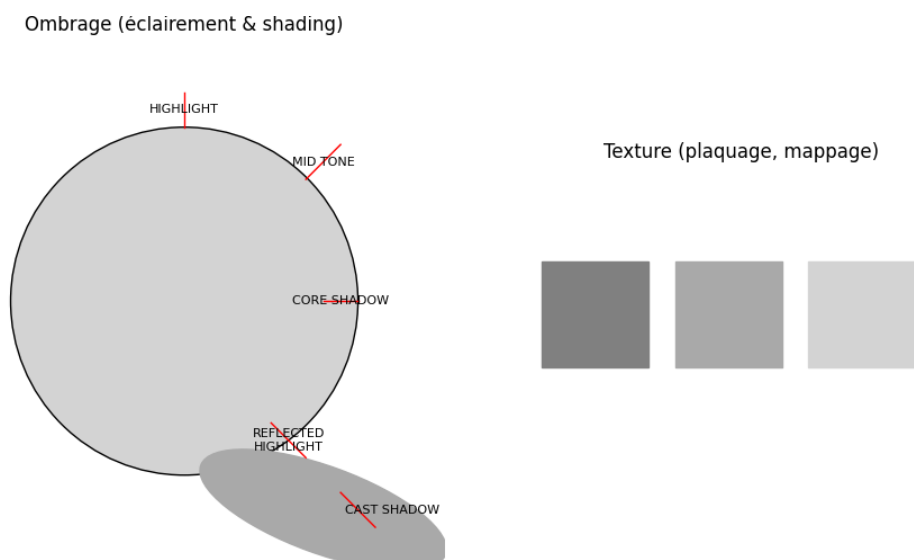


Figure 2: Illustration de l'Ombrage et de la Texture.

2 Définitions

Definition 2.1 (Illumination). Transport du flux lumineux direct ou indirect depuis les sources lumineuses vers les surfaces de la scène. On distingue les modèles locaux (interaction directe source-surface) des modèles globaux (prenant en compte les interactions entre objets, comme les ombres portées ou les réflexions indirectes).

Definition 2.2 (Éclairage). Calcul de l'intensité lumineuse en un point précis de la scène. Il modélise l'interaction entre une source lumineuse et le point éclairé.

Definition 2.3 (Ombrage (Shading)). Utilisation du modèle d'éclairage pour déterminer la couleur finale d'un pixel à l'écran, souvent par interpolation des valeurs calculées aux sommets des primitives.

3 Éclairement et Ombrage

Le calcul de l'éclairement et de l'ombrage en un point dépend de plusieurs facteurs :

- La position du point dans l'espace.
- L'orientation de la surface au niveau de ce point (définie par la normale à la surface).
- Les caractéristiques de la surface (comment elle diffuse ou réfléchit la lumière, sa transparence, sa couleur, sa texture...).
- Les sources de lumière présentes dans la scène (leur type : directionnelles, ponctuelles, etc., leur intensité, leur couleur).
- La position et l'orientation de la "caméra" ou du point de vue de l'observateur.

4 Sources de Lumières

4.1 Lumière ambiante

- Représente une lumière de fond qui éclaire toute la scène uniformément, simulant la lumière indirecte réfléchiée par l'environnement.
- Elle n'a pas de direction ni de position spécifique.
- Est uniquement caractérisée par son **intensité** (et sa couleur).

4.2 Sources ponctuelles

- Sources de lumière placées en un point précis de l'espace.
- Elles émettent de la lumière radialement dans toutes les directions (isotropique) ou dans des directions privilégiées (anisotropique).
- Caractérisées par leur **intensité**, leur **position** et une fonction d'**atténuation** ('falloff') qui décrit comment l'intensité diminue avec la distance.
- Lorsque la source ponctuelle prend du volume (n'est plus un point mathématique), elle devient une "source étendue".

4.3 Sources directionnelles

- Simulent une source de lumière très éloignée (comme le soleil).
- Éclairent la scène avec des rayons lumineux parallèles provenant tous d'une même direction.
- Caractérisées par leur **intensité** et leur **direction**. La position n'est pas pertinente.

4.4 Sources projecteur ou spot

- Sources de lumière qui émettent un cône de lumière dans une direction spécifique.
- Caractérisées par leur **position**, leur **direction**, un **angle d'ouverture** du cône et un **facteur de concentration** (comment l'intensité diminue du centre vers les bords du cône).

Note: Les diagrammes illustrant les sources lumineuses ne sont pas recréés ici, mais leur description est fournie ci-dessus.

5 Modèles d'Éclairage

Plusieurs modèles sont utilisés pour simuler l'interaction de la lumière avec les surfaces :

- Lumière émise
- Lumière ambiante
- Réflexion diffuse
- Réflexion spéculaire
- Brillance (associée à la réflexion spéculaire)

5.1 Lumière Émise

- Certains objets peuvent eux-mêmes émettre de la lumière (par exemple, une lampe, un écran).
- Dans les modèles locaux simples, on considère souvent que les objets ne sont pas intrinsèquement émetteurs de lumière. Ils ne font que réfléchir la lumière des sources externes.
- La lumière émise peut être considérée comme un niveau minimum d'éclairage propre à l'objet, indépendant des sources externes.

5.2 Lumière Ambiante

- Correspond au modèle d'éclairage le plus simple.
- On considère qu'il existe une source lumineuse virtuelle présente partout, éclairant de manière égale dans toutes les directions.
- Physiquement, cela approxime la lumière indirecte réfléchie par l'environnement global (par exemple, la lumière du ciel ou les réflexions multiples dans une pièce).
- Ce modèle assure qu'aucun objet, même non directement éclairé par une source ponctuelle ou directionnelle, ne soit complètement noir. Il fournit un niveau minimum d'éclairage appliqué sur tous les objets.
- On définit l'intensité résultant de la lumière ambiante réfléchie par une surface (I_p) comme :

$$I_p = p_a \cdot I_a$$

où :

- I_a est l'intensité de la lumière ambiante incidente.
- p_a est le coefficient de réflexion ambiante de la surface (une propriété du matériau, comprise entre 0 et 1, souvent de la même couleur que la réflexion diffuse).
- I_p est l'intensité (couleur) résultante réfléchie par la surface due à la composante ambiante.
- Cette intensité I_p est constante sur toute la surface de l'objet, quelle que soit son orientation ou la position de l'observateur.

Propriétés :

- Ne permet pas de percevoir la forme 3D des objets car l'éclairage est uniforme.
- Modélise simplement l'interréflexion globale entre toutes les surfaces d'une scène.
- Évite qu'un objet dans l'ombre soit complètement noir.

Note: L'illustration des théières avec augmentation de p_a montre simplement des objets devenant uniformément plus clairs.

5.3 Réflexion Diffuse

- Simule la réflexion de la lumière sur des surfaces mates ou rugueuses (non brillantes).
- Si on considère une surface **Lambertienne** (idéalement diffuse), la lumière incidente est réfléchie de manière égale dans toutes les directions, indépendamment de la position de l'observateur.
- L'intensité de la lumière réfléchie dépend de :
 - I_l : L'intensité de la source lumineuse incidente.
 - θ : L'angle entre le vecteur pointant vers la source lumineuse (\vec{L}) et la normale à la surface (\vec{N}) au point considéré.
 - p_d : Le coefficient de réflexion diffuse de la surface (propriété du matériau, $0 \leq p_d \leq 1$).
- Principe physique : Une source lumineuse émet une certaine énergie par unité de surface. Selon l'angle d'incidence des rayons lumineux, cette énergie se répartit sur une surface plus ou moins grande de l'objet. Si les rayons sont perpendiculaires à la surface ($\theta = 0$), l'énergie est concentrée sur la plus petite surface possible, résultant en une intensité réfléchie maximale. Plus l'angle θ augmente, plus l'énergie se répartit sur une grande surface, et l'intensité réfléchie diminue. Cette dépendance est modélisée par le terme $\cos(\theta)$.
- La formule pour l'intensité diffuse réfléchie (I_p) est :

$$I_p = p_d \cdot I_l \cdot \cos(\theta)$$

ou en utilisant le produit scalaire des vecteurs normalisés :

$$I_p = p_d \cdot I_l \cdot (\vec{N} \cdot \vec{L})$$

Note : Si $\vec{N} \cdot \vec{L} < 0$, cela signifie que la lumière vient de derrière la surface, donc l'intensité diffuse est nulle ($\cos(\theta)$ est pris comme $\max(0, \cos(\theta))$).

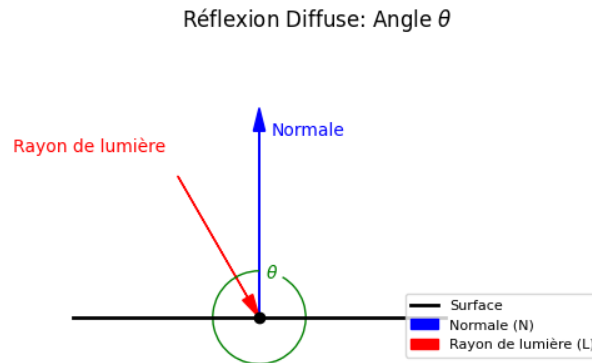


Figure 3: Angle θ utilisé dans le calcul de la réflexion diffuse.

Note: L'illustration des théières avec $p_a = 0$ et augmentation de p_d montre des objets dont la luminosité dépend de l'orientation par rapport à la lumière, avec des surfaces face à la lumière plus claires et des surfaces perpendiculaires ou orientées à l'opposé plus sombres.

5.4 Modèle Diffuse + Ambiante

On combine souvent les composantes ambiante et diffuse pour obtenir un éclairage plus réaliste :

$$I_{total} = I_{ambient} + I_{diffus} = p_a I_a + p_d I_l \max(0, \vec{N} \cdot \vec{L})$$

Note: Le diagramme de grille de théières montre l'effet combiné: l'augmentation de p_a (verticalement) augmente la luminosité globale, tandis que l'augmentation de p_d (horizontalement) augmente le contraste dû à l'orientation.

5.5 Réflexion Spéculaire

- Simule la réflexion sur des surfaces brillantes ou polies (effet "miroir").
- Contrairement à la réflexion diffuse, la lumière est réfléchi préférentiellement dans une certaine direction.
- **Modèle de Phong [1973]:** Le modèle le plus connu. Il fait intervenir la position de l'observateur. Les rayons lumineux sont réfléchis de manière symétrique par rapport à la normale à la surface.
- L'intensité spéculaire observée dépend de :
 - I_l : L'intensité de la source lumineuse incidente.
 - θ' : L'angle entre le rayon réfléchi idéal (\vec{R}) et le vecteur pointant vers le point de vue (\vec{V}).
 - p_s : Le coefficient de réflexion spéculaire de la surface ($0 \leq p_s \leq 1$).
 - n : L'exposant de **rugosité** (ou brillance, *shininess*), qui contrôle la netteté du reflet.
- Le rayon réfléchi \vec{R} est calculé par : $\vec{R} = 2(\vec{N} \cdot \vec{L})\vec{N} - \vec{L}$ (avec \vec{L} pointant *vers* la source).
- L'intensité spéculaire est maximale lorsque l'observateur regarde exactement dans la direction du rayon réfléchi ($\theta' = 0$). Elle diminue rapidement lorsque l'angle θ' augmente. Cette diminution est modélisée par $\cos^n(\theta')$.
- La formule pour l'intensité spéculaire de Phong (I_s) est :

$$I_s = p_s \cdot I_l \cdot (\cos(\theta'))^n$$

ou en utilisant le produit scalaire des vecteurs normalisés :

$$I_s = p_s \cdot I_l \cdot (\vec{R} \cdot \vec{V})^n$$

Note : Si $\vec{R} \cdot \vec{V} < 0$, l'intensité spéculaire est nulle ($\cos(\theta')$ est pris comme $\max(0, \cos(\theta'))$).

- L'exposant n (rugosité / shininess) :
 - Une valeur **élevée** de n (ex: 100, voire ∞ ou 1024 pour un miroir parfait) produit une petite tâche spéculaire très intense (surface très lisse).
 - Une valeur **faible** de n (ex: 1 à 10) produit une tâche spéculaire large et diffuse (surface plus rugueuse).

Réflexion Spéculaire (Phong): Angle θ'

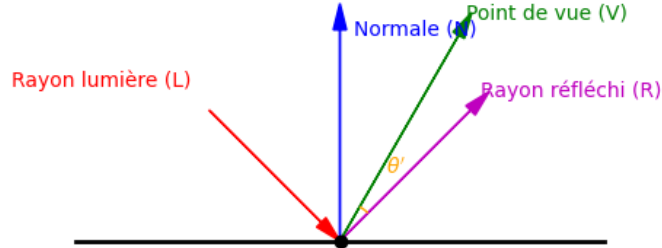


Figure 4: Angles et vecteurs pour la réflexion spéculaire de Phong.

- **Modèle Blinn-Phong [1977]:** Une version simplifiée et souvent plus rapide à calculer.
- Au lieu de calculer le rayon réfléchi \vec{R} , on calcule le vecteur **demi-angle** (halfway vector) \vec{H} , qui est à mi-chemin entre le vecteur lumière \vec{L} et le vecteur vue \vec{V} .

$$\vec{H} = \frac{\vec{L} + \vec{V}}{\|\vec{L} + \vec{V}\|}$$

- L'intensité spéculaire dépend alors de l'angle θ'' entre la normale \vec{N} et le vecteur demi-angle \vec{H} .
- La formule devient :

$$I_s = p_s \cdot I_l \cdot (\cos(\theta''))^n$$

ou en utilisant le produit scalaire des vecteurs normalisés :

$$I_s = p_s \cdot I_l \cdot (\vec{N} \cdot \vec{H})^n$$

Note : Si $\vec{N} \cdot \vec{H} < 0$, l'intensité spéculaire est nulle ($\cos(\theta'')$ est pris comme $\max(0, \cos(\theta''))$).

- Ce modèle est souvent préféré car \vec{H} est plus simple à calculer que \vec{R} , surtout quand \vec{V} et \vec{L} sont constants (vue ou source à l'infini). Les résultats visuels sont très similaires à ceux de Phong, bien que l'exposant n puisse nécessiter un ajustement pour obtenir le même aspect.

Réflexion Spéculaire (Blinn-Phong): Angle θ''

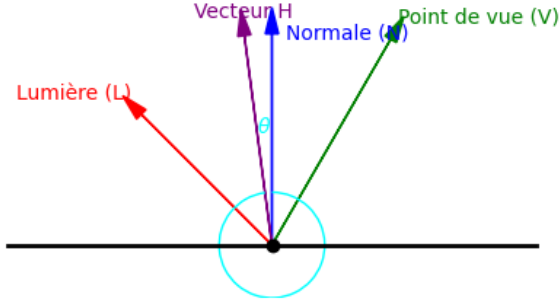


Figure 5: Vecteur demi-angle \vec{H} et angle θ'' pour Blinn-Phong.

Note: Les sphères comparant Blinn-Phong et Phong montrent des résultats visuels très similaires.

5.6 Modèle d'Éclairage Complet (Phong ou Blinn-Phong)

Le modèle d'illumination complet combine les composantes ambiante, diffuse et spéculaire pour calculer l'intensité finale (I) en un point P :

$$I(P) = I_{\text{ambient}} + I_{\text{diffus}} + I_{\text{speculaire}}$$

En utilisant les notations précédentes et en introduisant un facteur d'atténuation F_d qui peut dépendre de la distance à la source lumineuse (pour les sources ponctuelles/spots), l'équation pour une seule source lumineuse l devient :

$$I(P) = p_a I_a + F_{d,l} \cdot (p_d \cdot I_l \cdot \max(0, \vec{N} \cdot \vec{L}_l) + p_s \cdot I_l \cdot \max(0, \vec{N} \cdot \vec{H}_l)^n)$$

(Ici, la version Blinn-Phong est utilisée pour le terme spéculaire).

Note: Le diagramme de grille de théières montre l'effet combiné : l'augmentation de p_s (marquée 'ps' ou 'n' sur le slide) ajoute ou renforce les reflets brillants.

Note: Les images des théières de la page 7 illustrent le résultat visuel de l'application du modèle complet avec différents paramètres.

5.7 Modèle coloré

Pour obtenir des couleurs, les calculs sont effectués séparément pour chaque composante de couleur (Rouge, Vert, Bleu - RVB) :

- L'intensité de la lumière ambiante I_a et de chaque source I_l ont des composantes (I_{ar}, I_{ag}, I_{ab}) et (I_{lr}, I_{lg}, I_{lb}).
- Les coefficients de réflexion p_a, p_d, p_s sont également définis par composante : (p_{ar}, p_{ag}, p_{ab}), (p_{dr}, p_{dg}, p_{db}), (p_{sr}, p_{sg}, p_{sb}). Ces coefficients représentent la couleur intrinsèque du matériau.

- L'équation complète est appliquée pour chaque canal R, V, B :

$$\begin{aligned}
I_r &= p_{ar}I_{ar} + \sum_l F_{d,l}(p_{dr}I_{lr}(\vec{N} \cdot \vec{L}_l) + p_{sr}I_{lr}(\vec{N} \cdot \vec{H}_l)^n) \\
I_g &= p_{ag}I_{ag} + \sum_l F_{d,l}(p_{dg}I_{lg}(\vec{N} \cdot \vec{L}_l) + p_{sg}I_{lg}(\vec{N} \cdot \vec{H}_l)^n) \\
I_b &= p_{ab}I_{ab} + \sum_l F_{d,l}(p_{db}I_{lb}(\vec{N} \cdot \vec{L}_l) + p_{sb}I_{lb}(\vec{N} \cdot \vec{H}_l)^n)
\end{aligned}$$

(Les termes $\max(0, \cdot)$ sont omis pour la lisibilité).

Si plusieurs sources lumineuses sont présentes, leurs contributions (diffuse et spéculaire) sont sommées. La composante ambiante est généralement ajoutée une seule fois.

5.8 Transparence

Pour gérer la transparence, un paramètre t (souvent appelé alpha, α) est introduit ($0 \leq t \leq 1$, où $t = 1$ est opaque et $t = 0$ est transparent). La couleur finale I d'un fragment semi-transparent est calculée en combinant sa propre couleur calculée $I(P)$ avec la couleur $I(\text{derrière } P)$ de ce qui se trouve derrière lui :

$$I = t \cdot I(P) + (1 - t) \cdot I(\text{derrière } P)$$

5.9 Halo

Le halo simule un effet de lueur autour des objets lumineux ou réfléchissants. La couleur et l'intensité du halo dépendent souvent de l'épaisseur de l'objet traversé ou de l'intensité de la source.

6 Modèles d'Ombrage (Shading)

L'ombrage (*shading*) est la technique utilisée pour appliquer les modèles d'éclairage aux surfaces des objets (souvent représentés par des maillages de polygones, typiquement des triangles) afin de déterminer la couleur de chaque pixel.

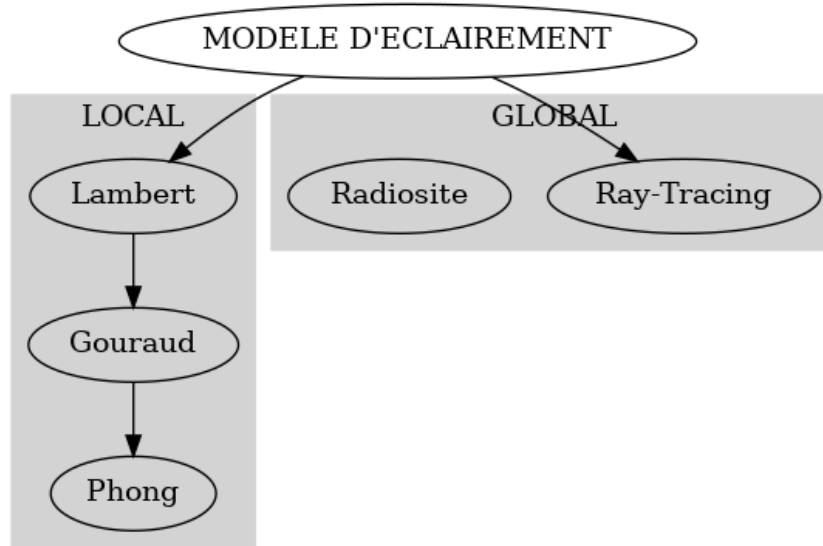


Figure 6: Classification des modèles d'ombrage.

- **Modèles locaux** : La luminance à la surface d'un objet est calculée en considérant l'objet comme isolé. On utilise uniquement les paramètres de l'objet lui-même et les paramètres des sources de lumière directes. (Ex: Lambert, Gouraud, Phong).
- **Modèles globaux** : La luminance est calculée en tenant compte des interactions avec tous les objets de la scène (ombres portées, réflexions indirectes, etc.). (Ex: Ray Tracing, Radiosité).

Nous nous concentrons ici sur les modèles locaux.

6.1 Ombrage de Lambert (Plat / Constant)

- La méthode d'ombrage la plus simple pour les facettes polygonales.
- Une seule valeur d'illumination est calculée pour l'ensemble de la facette et appliquée uniformément à tous ses pixels.
- Le calcul est typiquement fait au centre de la facette, en utilisant la normale géométrique de la facette elle-même (qui est constante sur toute la facette).
- Formule (pour la composante diffuse) : $I_{facette} = p_d \cdot I_l \cdot \max(0, \vec{N}_{facette} \cdot \vec{L})$
- **Problème** : Crée des discontinuités d'intensité visibles aux arêtes entre les facettes, donnant un aspect facetté. L'œil humain a tendance à exagérer ces changements brusques d'intensité (effet de *Mach Banding*).
- **Solution** : Utiliser une interpolation pour lisser les transitions (voir Gouraud et Phong).

Note: L'illustration des sphères montre clairement l'aspect facetté avec 16, 32 et 64 facettes. Note: Le diagramme d'intensité illustre la discontinuité perçue (marches d'escalier).

6.2 Ombrage de Gouraud [1971]

- Développé par Henri Gouraud pour éliminer les discontinuités de l'ombrage plat.
- Principe : Interpoler les valeurs d'intensité (couleur) aux sommets de la facette.
- Étapes :
 1. **Calculer les normales aux sommets** : Pour chaque sommet, calculer une normale qui représente la moyenne de l'orientation de la surface autour de ce sommet. Cela se fait généralement en moyennant les normales des facettes qui partagent ce sommet. $N_s = \frac{\sum N_i}{\|\sum N_i\|}$, où N_i sont les normales des faces incidentes au sommet s .
 2. **Calculer l'intensité aux sommets** : Appliquer le modèle d'éclairage (par exemple, Ambient + Diffus + Spéculaire) à chaque sommet en utilisant sa normale calculée (N_s) pour obtenir une intensité (I_{s1}, I_{s2}, I_{s3} pour un triangle). $I_s = p_a I_a + p_d I_l (\vec{N}_s \cdot \vec{L}) + p_s I_l (\vec{N}_s \cdot \vec{H})^n$.
 3. **Interpoler les intensités à l'intérieur de la facette** : Pour chaque pixel à l'intérieur de la facette, interpoler linéairement (ou bilinéairement) les intensités calculées aux sommets. Ceci est souvent fait par scan-line :
 - Interpoler les intensités le long des arêtes gauche et droite de la facette pour la ligne de balayage actuelle (y) pour obtenir I_{gauche} et I_{droite} .

$$I_{gauche} = I_{s1} \frac{y - y_2}{y_1 - y_2} + I_{s2} \frac{y - y_1}{y_2 - y_1}$$

(Exemple pour l'arête S1-S2, adaptation pour les autres arêtes).

- Interpoler horizontalement entre I_{gauche} et I_{droite} pour la coordonnée x du pixel.

$$I_{pixel}(x, y) = I_{gauche} \frac{x - x_{droite}}{x_{gauche} - x_{droite}} + I_{droite} \frac{x - x_{gauche}}{x_{droite} - x_{gauche}}$$

(Simplifié, utilise souvent les coordonnées barycentriques).

- **Caractéristiques :**

- Produit un aspect lisse et continu.
- Efficace en termes de calcul (calcul du modèle d'éclairage uniquement aux sommets).
- Facile à implémenter avec des algorithmes de remplissage de polygones comme le Z-buffer (scan-line).
- **Inconvénients :** N'est qu'une approximation. Peut manquer ou déformer les reflets spéculaires s'ils se produisent au milieu d'une facette (car l'interpolation se fait sur les intensités finales). Peut produire des artefacts visuels (comme des bandes de Mach si la tessellation n'est pas assez fine).

Note: Le pseudo-code pour l'interpolation de Gouraud est omis mais les étapes sont décrites ci-dessus.

6.3 Ombrage de Phong [1973]

- Développé pour améliorer la gestion des reflets spéculaires par rapport à Gouraud.
- Principe : Interpoler les vecteurs *normaux* aux sommets, puis appliquer le modèle d'éclairage complet à chaque pixel en utilisant la normale interpolée.
- Étapes :
 1. **Calculer les normales aux sommets :** Identique à Gouraud (N_{s1}, N_{s2}, N_{s3}).
 2. **Interpoler les normales à l'intérieur de la facette :** Pour chaque pixel, interpoler linéairement (ou bilinéairement) les vecteurs normaux des sommets pour obtenir une normale interpolée \vec{N}_{pixel} . L'interpolation se fait de la même manière que pour les intensités dans Gouraud (par scan-line ou coordonnées barycentriques).

$$\vec{N}_{pixel}(x, y) = Interpolation(\vec{N}_{s1}, \vec{N}_{s2}, \vec{N}_{s3})$$

(Important: le vecteur normal interpolé doit être *renormalisé* après interpolation).

3. **Calculer l'intensité au pixel :** Appliquer le modèle d'éclairage complet (Ambiant + Diffus + Spéculaire) pour le pixel courant en utilisant la normale interpolée et renormalisée \vec{N}_{pixel} .

$$I_{pixel} = p_a I_a + p_d I_l(\vec{N}_{pixel} \cdot \vec{L}) + p_s I_l(\vec{N}_{pixel} \cdot \vec{H})^n$$

- **Caractéristiques :**

- Produit des reflets spéculaires plus précis et réalistes que Gouraud, même s'ils apparaissent au milieu des facettes.
- Est généralement considéré comme donnant un meilleur rendu visuel que Gouraud, même pour des modèles sans spécularité forte.
- **Inconvénients :** Nettement plus coûteux en calcul que Gouraud, car le modèle d'éclairage complet (incluant potentiellement des racines carrées pour la normalisation et des exponentiations pour le spéculaire) doit être évalué pour chaque pixel, et non juste aux sommets.

Note: Les comparaisons visuelles (sphères, visages) montrent que Lambert est facetté, Gouraud est lisse mais avec des reflets spéculaires diffus/manquants, et Phong est lisse avec des reflets nets et bien placés.

6.4 Ajout des Couleurs et Plusieurs Sources

Comme pour les modèles d'éclairage, l'ombrage est appliqué par composante de couleur (RVB). Pour gérer plusieurs sources lumineuses, les contributions diffuses et spéculaires de chaque source sont additionnées à la composante ambiante unique.

$$I_{pixel} = \text{Ambiant} + \sum_{l \in \text{lumières}} (\text{Diffus}_l + \text{Spéculaire}_l)$$

où Ambiant , Diffus_l , Spéculaire_l sont calculés en utilisant les normales appropriées (facette pour Lambert, sommet pour Gouraud, pixel interpolé pour Phong) et les propriétés de la lumière l .

7 Interpolation Linéaire et Coordonnées Barycentriques

L'interpolation est fondamentale pour les ombrages de Gouraud et Phong.

7.1 Interpolation Linéaire en 1D

Pour interpoler une valeur f entre deux points x_i et x_j où les valeurs sont f_i et f_j , on utilise un paramètre $t \in [0, 1]$:

$$f(t) = f_i + t(f_j - f_i) = (1 - t)f_i + tf_j$$

où $t = (x - x_i)/(x_j - x_i)$. Cela peut être vu comme une combinaison linéaire de deux fonctions de base, $(1 - t)$ et t .

7.2 Interpolation Linéaire en 2D (pour les triangles)

Pour interpoler une valeur (intensité, normale, coordonnée de texture, etc.) f à l'intérieur d'un triangle défini par les sommets p_i, p_j, p_k avec les valeurs f_i, f_j, f_k , on utilise les **coordonnées barycentriques**.

Un point p à l'intérieur du triangle peut s'écrire comme une combinaison affine des sommets : $p = \alpha p_i + \beta p_j + \gamma p_k$, avec $\alpha, \beta, \gamma \geq 0$ et $\alpha + \beta + \gamma = 1$. Les coefficients (α, β, γ) sont les coordonnées barycentriques de p . Ils peuvent être calculés comme des rapports d'aires :

$$\begin{aligned}\alpha &= \frac{\text{area}(p, p_j, p_k)}{\text{area}(p_i, p_j, p_k)} \\ \beta &= \frac{\text{area}(p, p_i, p_k)}{\text{area}(p_i, p_j, p_k)} \\ \gamma &= \frac{\text{area}(p, p_i, p_j)}{\text{area}(p_i, p_j, p_k)} \quad (= 1 - \alpha - \beta)\end{aligned}$$

La valeur interpolée $f(p)$ est alors :

$$f(p) = \alpha f_i + \beta f_j + \gamma f_k$$

Ces coordonnées sont calculées lors de la rasterisation (remplissage) du triangle et permettent d'interpoler n'importe quel attribut défini aux sommets.

8 Interpolation en Projection Perspective

8.1 Le Problème

L'interpolation barycentrique directe des attributs dans l'espace écran 2D après une projection perspective ne correspond pas à une interpolation linéaire dans l'espace 3D d'origine. Cela est dû à la division par la profondeur (ou coordonnée w) lors de la projection. Si on interpole linéairement en 2D des attributs (comme les coordonnées de texture), on obtient des déformations incorrectes (non linéaires en 3D).

8.2 Interpolation avec Correction de Perspective

Pour obtenir une interpolation correcte en perspective, il faut interpoler les attributs d'une manière qui tienne compte de la profondeur.

- Soit q l'attribut à interpoler (ex: coordonnée de texture u ou v , couleur R, G ou B) et z la profondeur (distance à la caméra) au sommet.
- Au lieu d'interpoler q directement, on calcule $P = q/z$ et $Z = 1/z$ pour chaque sommet.
- On interpole P et Z linéairement dans l'espace écran 2D en utilisant les coordonnées barycentriques (α, β, γ) :

$$\begin{aligned}P_{interp} &= \alpha P_i + \beta P_j + \gamma P_k \\Z_{interp} &= \alpha Z_i + \beta Z_j + \gamma Z_k\end{aligned}$$

- La valeur finale de l'attribut interpolé correctement en perspective pour le pixel est obtenue en divisant P_{interp} par Z_{interp} :

$$q_{pixel} = \frac{P_{interp}}{Z_{interp}}$$

Cette méthode assure que l'interpolation est linéaire dans l'espace 3D. Elle est essentielle pour obtenir des textures et des couleurs correctes sur des surfaces inclinées par rapport à l'observateur.

Note: L'illustration du quadrilatère divisé montre la discontinuité qui apparaît si la correction de perspective n'est pas utilisée lors de l'interpolation (par exemple, de coordonnées de texture).

9 Textures (Plaquage et Mappage)

Le **Texture Mapping** (plaquage de texture) consiste à appliquer une image (la texture) sur la surface d'un objet 3D pour ajouter des détails visuels sans augmenter la complexité géométrique du modèle.

9.1 Motivation

Calculer l'éclairage par sommet et interpoler (Gouraud/Phong) peut nécessiter une géométrie très fine pour représenter des détails fins. Le plaquage de texture est une alternative plus efficace : on calcule un éclairage plus simple sur une géométrie moins complexe, puis on module le résultat avec une image de texture détaillée.

9.2 Types de Textures

Il existe de nombreux types de textures utilisés pour différents effets :

- **Couleurs (Colour map / Diffuse map / Albedo)** : Définit la couleur de base de la surface.
- **Transparence (Alpha map)** : Contrôle l'opacité de la surface (souvent dans le canal alpha de la texture de couleur).
- **Relief (Bump maps, Normal maps, Displacement maps)** : Simulent des détails de relief en modifiant la normale perçue (Normal map) ou en déplaçant réellement les sommets (Displacement map), sans ajouter de polygones.
- **Spéculaires (Specular map, Gloss map, Roughness map, Metallic map)** : Contrôlent comment la surface réfléchit la lumière spéculaire (intensité, couleur, netteté du reflet, caractère métallique).
- **Environnement (Environment map, Cube map)** : Utilisées pour simuler des réflexions de l'environnement sur des surfaces réfléchissantes.
- **Lumière (Light map)** : Stocke un éclairage précalculé (souvent l'illumination globale indirecte) qui est ensuite ajouté à l'éclairage dynamique.
- **Autres** : Occlusion ambiante, émissivité, épaisseur (pour le subsurface scattering), etc.

9.3 Attributs des Textures et Combinaison

Les textures peuvent être combinées pour créer des effets complexes :

- **Diffuse + Spéculaire** : Une texture de couleur (diffuse) et une texture contrôlant la brillance (spéculaire).
- **Diffuse + Transparence** : Utilisation du canal alpha pour rendre certaines parties invisibles ou semi-transparentes.
- **Illumination précalculée (Light map)** : Multipliée ou ajoutée à la couleur de base pour simuler un éclairage statique complexe.
- **Simulation d'environnement** : Une texture cubique (Cube map) représentant l'environnement autour de l'objet est utilisée pour calculer les réflexions.
- **Normal Mapping** : Une texture spéciale (Normal map) stocke des vecteurs normaux. Lors du calcul de l'éclairage, la normale lue dans cette texture est utilisée à la place de la normale géométrique, créant une illusion de relief détaillé sur une surface plane.
- **Displacement Mapping** : Similaire au Normal Mapping, mais les valeurs de la texture sont utilisées pour déplacer réellement les sommets de la géométrie le long de leur normale, créant un relief réel (plus coûteux).

9.4 Fonction de Plaquage (Mapping)

Le processus consiste à établir une correspondance entre les points de la surface 3D de l'objet $P(x, y, z)$ et les points (coordonnées) (s, t) ou (u, v) dans l'image de texture 2D $I(s, t)$. Cette correspondance est la fonction de plaquage $f : P(x, y, z) \rightarrow (s, t)$.

Coordonnées de Texture (UV Mapping) : La méthode la plus courante consiste à assigner des coordonnées de texture (u, v) à chaque sommet du modèle 3D. Ces coordonnées (u, v) se situent généralement dans l'intervalle $[0, 1]$ et indiquent quelle partie de l'image de texture correspond à ce sommet. Lors de la rasterisation, ces coordonnées (u, v) sont interpolées sur la surface du polygone (en utilisant l'interpolation avec correction de perspective !) pour chaque pixel. La coordonnée (u, v) interpolée est ensuite utilisée pour lire la couleur (ou autre valeur) dans l'image de texture.

Méthodes de projection pour générer les UV : Pour les objets simples, les coordonnées UV peuvent être générées par des projections mathématiques :

- **Planaire** : Projection sur un plan. $f(x, y, z) = (k_x x + c_x, k_y y + c_y)$. Convient aux objets plats.
- **Cylindrique** : Enroulement autour d'un cylindre. Les coordonnées sont basées sur l'angle θ autour de l'axe du cylindre et la hauteur y . $f(\theta, y) = (\theta/(2\pi), y/H)$, où $\theta = \text{atan2}(x, z)$.
- **Sphérique** : Basé sur les angles de longitude (θ) et latitude (ϕ). $f(\theta, \phi) = (\theta/(2\pi), \phi/\pi)$, où $\theta = \text{atan2}(x, z)$, $\phi = \text{asin}(y/R)$.

Pour les modèles complexes, les coordonnées UV sont souvent créées manuellement par un artiste 3D dans un processus appelé "UV unwrapping" (dépliage UV).

9.5 Interprétation des Coordonnées Hors $[0, 1]$

Que faire si les coordonnées (u, v) interpolées tombent en dehors de l'intervalle $[0, 1]$? Plusieurs modes d'adressage de texture existent :

- **GL_CLAMP_TO_EDGE** : La coordonnée est limitée à l'intervalle $[0, 1]$. La couleur du bord de la texture est répétée.
- **GL_REPEAT** : La partie entière de la coordonnée est ignorée, seule la partie fractionnaire est utilisée. La texture se répète.
- **GL_MIRRORED_REPEAT** : Similaire à REPEAT, mais la texture est répétée en miroir à chaque fois.

9.6 Problèmes et Limites

- **Distorsion** : Selon la méthode de plaquage UV, la texture peut apparaître étirée ou compressée sur certaines parties du modèle. Un bon dépliage UV vise à minimiser ces distorsions.
- **Crénelage (Aliasing)** : Apparaît lorsque la résolution de la texture ne correspond pas bien à la taille à laquelle elle est affichée à l'écran.

9.7 Magnification vs. Minification

- **Magnification (Agrandissement)** : La caméra est proche de l'objet texturé. Un pixel à l'écran correspond à une petite zone (moins d'un texel) dans la texture.
 - *Problème* : Peut rendre la texture pixellisée si on prend simplement le texel le plus proche (*Nearest neighbor filtering*).
 - *Solution* : Interpolation bilinéaire (moyenne pondérée des 4 texels voisins) pour un résultat plus lisse.
- **Minification (Réduction)** : La caméra est loin de l'objet texturé. Un pixel à l'écran correspond à une grande zone (plusieurs texels) dans la texture.
 - *Problème* : Si on échantillonne juste un texel au centre de la zone couverte par le pixel, on manque beaucoup d'informations, ce qui conduit à des artefacts scintillants (aliasing) et des motifs moirés.
 - *Solution* : Il faut calculer une couleur moyenne de tous les texels couverts par le pixel. C'est coûteux à faire dynamiquement. Une solution est le **MIP-Mapping**.

9.8 Aliasing et MIP-Mapping

Le **MIP-Mapping** est une technique de préfiltrage pour lutter contre l'aliasing de minification.

- **Idée** : Créer une pyramide d'images pré-filtrées de la texture originale, où chaque niveau est une version réduite de moitié et moyennée du niveau précédent. Le niveau 0 est la texture originale (X_0, Y_0) . Le niveau 1 est $(X_0/2, Y_0/2)$, le niveau 2 est $(X_0/4, Y_0/4)$, etc., jusqu'à une texture 1x1 pixel. (MIP vient de *multum in parvo*, "beaucoup de choses dans un petit espace").
- **Utilisation** : En fonction de la distance de l'objet ou de la taille projetée de la texture à l'écran, le système graphique choisit le niveau de MIP-map le plus approprié (celui où un pixel écran correspond à environ un texel de ce niveau de MIP-map).
- **Filtrage inter-niveaux** : On peut encore améliorer la qualité en interpolant linéairement entre les deux niveaux de MIP-map les plus proches (*filtrage trinéaire*).
- **Modes OpenGL** :
 - `'GL_NEAREST'` / `'GL_LINEAR'` : Pas de MIP-Mapping (bon pour magnification). `'GL_NEAREST_MIPMAP_NEAREST'` / `'GL_LINEAR_MIPMAP_NEAREST'` : Choisit le meilleur niveau MIP, prend le texelle plus proche.
 - `'GL_LINEAR_MIPMAP_NEAREST'` / `'GL_NEAREST_MIPMAP_LINEAR'` : Choisit le meilleur niveau MIP, interpole bilinéairement dans ce niveau. `'GL_NEAREST_MIPMAP_LINEAR'` / `'GL_LINEAR_MIPMAP_LINEAR'` : Interpole linéairement entre les deux niveaux MIP.
 - `'GL_LINEAR_MIPMAP_LINEAR'` : Interpole bilinéairement dans deux niveaux MIP et interpole linéairement entre les deux niveaux MIP.

10 Ray Tracing

Le Ray Tracing (lancer de rayons) est une technique d'illumination globale alternative au pipeline de rasterisation standard.

10.1 Introduction

- Approche basée sur le pixel (*image-order rendering*) et non sur l'objet (*object-order rendering* comme la rasterisation).
- Principe "inverse-mapping" : Au lieu de projeter les objets vers l'écran, on lance des rayons depuis la caméra (l'œil) à travers chaque pixel de l'écran et on détermine ce que ce rayon rencontre dans la scène.
- Étapes de base :
 1. Pour chaque pixel de l'écran, construire un rayon partant de la caméra et passant par ce pixel.
 2. Trouver la première intersection de ce rayon avec un objet de la scène.
 3. Si une intersection est trouvée, déterminer la couleur en ce point d'intersection.

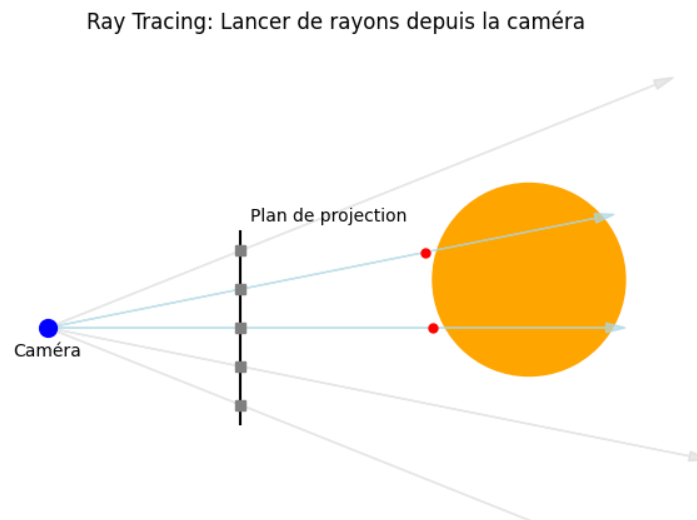


Figure 7: Principe du lancer de rayons.

10.2 Calcul de la Couleur au Point d'Intersection

Pour déterminer la couleur au point d'intersection, on applique un modèle d'éclairage (similaire à Phong), mais on peut aussi simuler des phénomènes globaux :

- **Ombres** : Depuis le point d'intersection, tracer un rayon ("rayon d'ombre") vers chaque source lumineuse. Si ce rayon rencontre un autre objet avant d'atteindre la source, le point est dans l'ombre de cette source (sa contribution directe est nulle).
- **Réflexion** : Si la surface est réfléchissante, lancer un nouveau rayon dans la direction de la réflexion (calculée comme dans Phong : $\vec{R} = 2(\vec{N} \cdot \vec{L})\vec{N} - \vec{L}$, où \vec{L} est ici la direction opposée au rayon incident). La couleur obtenue par ce rayon réfléchi est ajoutée à la couleur du point d'intersection, pondérée par le coefficient de réflexion de la surface.
- **Réfraction** : Si la surface est transparente ou translucide, lancer un nouveau rayon dans la direction de la réfraction (calculée par la loi de Snell-Descartes). La couleur obtenue par ce rayon réfracté est ajoutée, pondérée par le coefficient de transparence/réfraction.

10.3 Ray Tracing Récursif

Les rayons de réflexion et de réfraction peuvent eux-mêmes intersecter d'autres objets, générant à leur tour de nouveaux rayons d'ombre, de réflexion et de réfraction. Ce processus récursif permet de simuler des réflexions multiples, des réfractions complexes et des ombres douces (si on utilise des sources étendues). La récursion s'arrête après un certain nombre de rebonds ou lorsque l'intensité du rayon devient négligeable.

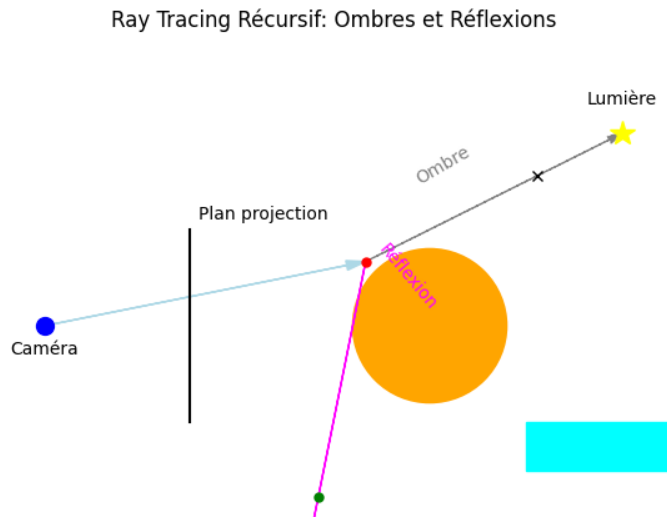


Figure 8: Illustration du lancer de rayons récursif avec ombres et réflexions.

Le Ray Tracing est capable de produire des images très réalistes mais est traditionnellement beaucoup plus coûteux en calcul que la rasterisation, bien que les accélérations matérielles récentes (RTX) le rendent de plus en plus viable pour le temps réel.