

Rendu

Partie 1.1

Modèle théorique

Partie 1.2

État de l'art

Partie 1.3

Éclairage ambiant

1.3.1

De la physique aux mathématiques

Le calcul de l'énergie reçu en un point de l'objet revient à intégrer le flux lumineux sur l'ensemble des direction visibles.

$$\mathcal{E}(p, \vec{n}) = \frac{1}{\pi} \int_{\mathcal{H}^2(\vec{n})} \mathcal{L}(p, \vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega} \quad (1.3.1.1)$$

L'intégrale selon $\int_{\mathcal{H}(\vec{n})} d\omega$ correspond à une intégrale suivant l'hémisphère visible et pondéré par l'angle solide en supposant qu'il n'y a pas d'occultation. Afin de modéliser les phénomènes d'auto-occultation on devra utiliser la formule suivante

$$\mathcal{E}(p, \vec{n}) = \frac{1}{\pi} \int_{\mathcal{V}(p, \vec{n})} \mathcal{L}(p, \vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega} \quad (1.3.1.2)$$

Ou $\mathcal{V}(p, \vec{n})$ est la restriction de l'hémisphère suivant le vecteur \vec{n} à l'espace effectivement visible depuis le point p (ce qui revient à considérer $\mathcal{H}(\vec{n})$ privée des direction correspondants à de l'auto-occultation).

Ici on fait d'abord la supposition que l'éclairement selon une direction $\vec{\omega}$ est indépendant du point considéré. Cette approximation, est nécessaire pour utiliser, sans reconstruction 3D complexe, l'envmap reconstruite dynamiquement.

On obtient donc l'équation

$$\mathcal{E}(p, \vec{n}) = \frac{1}{\pi} \int_{\mathcal{V}(p, \vec{n})} \mathcal{L}_{glob}(\vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega} \quad (1.3.1.3)$$

On fait alors la supposition que les phénomènes d'auto-occultation influencent de manière moyenne et globale l'énergie reçu en un point. Cela nous permet de revenir à une intégration sur tout le demi espace $\mathcal{H}^2(\vec{n})$ en ajoutant simplement un coefficient de $\mathcal{P}_{\mathcal{V}}(p)$ valant 1 en l'absence d'auto-occultation et 0 pour une auto-occultation totale.

Ce coefficient définit sur la surface de l'objet pourra être pré-calculé et fournit sous forme de texture.

$$\begin{aligned} \mathcal{P}_{\mathcal{V}}(p) &= \frac{\int_{\mathcal{V}(p, \vec{n}(p))} \vec{\omega} \cdot \vec{n} d\vec{\omega}}{\int_{\mathcal{H}(\vec{n}(p))} \vec{\omega} \cdot \vec{n} d\vec{\omega}} \\ &= \frac{1}{\pi} \int_{\mathcal{V}(p, \vec{n}(p))} \vec{\omega} \cdot \vec{n} d\vec{\omega} \end{aligned} \quad (1.3.1.4)$$

d'ou

$$\mathcal{E}(p, \vec{n}) = \frac{\mathcal{P}_{\mathcal{V}}(p)}{\pi} \int_{\mathcal{H}(p, \vec{n})} \mathcal{L}_{glob}(\vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega} \quad (1.3.1.5)$$

Enfin, on fera une dernière approximation suivant la méthode proposé dans [?] : on considère la fonction \mathcal{L}_{glob} (décrite par l'envmap) comme constante sur chaque face de l'envmap. Le niveau de mipmap le moins détaillé nous donne en effet une valeur moyenne pour la face considérée.

Si l'on oublie un instants les problèmes d'auto-occultation et que l'on s'intéresse a l'angle solide décrit par une face du cube centrée au point de vue, un rapide calcul permet d'évaluer l'angle solide décrit par une face complètement visible comme étant

$$\Omega_F = \iint_F \frac{1}{\|\vec{\omega}\|^3} d\omega = \frac{2\pi}{3} \quad (1.3.1.6)$$

L'angle solide sur une face partiellement visible (selon une demi sphère caractérisée par l'hyperplan de vecteur \vec{n}) est alors défini par

$$\Omega_F(\vec{n}) = \iint_F \frac{\text{HS}(\vec{\omega} \cdot \vec{n})}{\|\vec{\omega}\|^3} d\omega \quad (1.3.1.7)$$

Avec HS la fonction de HeavySide definie par :

$$\text{HS}(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

La figure 1.1 retrace l'évolution de $\Omega_F(\vec{n})$ selon l'angle θ pour des orientation de $\phi = 0$ et $\phi = \frac{\pi}{2}$

Les poids affectés aux valeurs de chaque face de l'envmap sont donc l'intégration du produit scalaire normalisé $\frac{\vec{\omega} \cdot \vec{n}}{\|\vec{\omega}\|}$ par rapport à l'angle solide sur la face considéré

$$\begin{aligned} W_F(\vec{n}) &= \iint_F \frac{\vec{\omega} \cdot \vec{n}}{\|\vec{\omega}\|} \times \frac{\text{HS}(\vec{\omega} \cdot \vec{n})}{\|\vec{\omega}\|^3} d\omega \\ &= \iint_F \frac{\vec{\omega} \cdot \vec{n} \times \text{HS}(\vec{\omega} \cdot \vec{n})}{\|\vec{\omega}\|^4} d\omega \end{aligned} \quad (1.3.1.8)$$

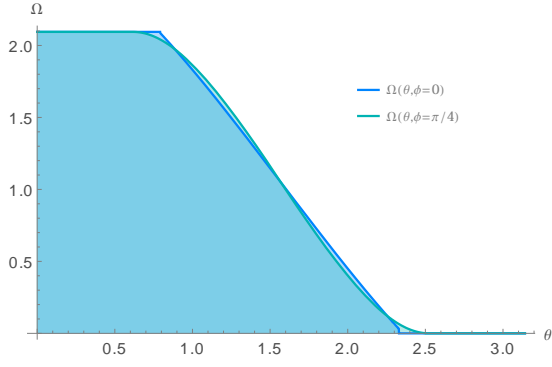
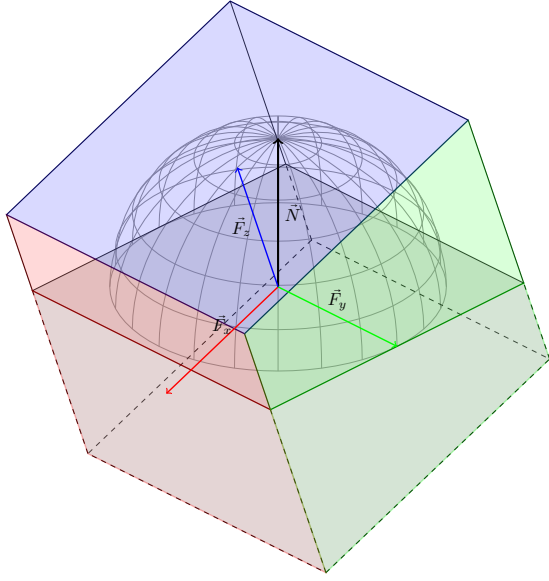

 FIGURE 1.1 – Évolution de $\Omega_F(\vec{n})$ selon l'orientation de \vec{n}


FIGURE 1.2 – Intégration de l'envmap par rapport à une facette

On notera qu'on a bien

$$\sum_{F \in F_{\text{Faces}}} \Omega(F) = 4\pi \quad (1.3.1.9a)$$

$$\forall \vec{n} \sum_{F \in F_{\text{Faces}}} \Omega(F, \vec{n}) = 2\pi \quad (1.3.1.9b)$$

$$\forall \vec{n} \sum_{F \in F_{\text{Faces}}} W_F(\vec{n}) = \pi \quad (1.3.1.9c)$$

Dès lors, l'approximation selon laquelle \mathcal{L} est constante sur chaque face nous donne

$$\begin{aligned} \mathcal{E}(p, \vec{n}) &= \frac{\mathcal{P}_V(p)}{\pi} \sum_{F \in F_{\text{Faces}}} \mathcal{L}_{\text{glob}}(\vec{F}) W_F(\vec{n}) \\ &= \frac{\mathcal{P}_V(p)}{\pi} \sum_{F \in F_{\text{Faces}}} \mathcal{L}_{\text{glob}}(\vec{F}) \iint_F \frac{\vec{\omega} \cdot \vec{n} \times \text{HS}(\vec{\omega} \cdot \vec{n})}{\|\vec{\omega}\|^4} d\omega \end{aligned} \quad (1.3.1.10)$$

1.3.2

Approximation numérique

En considérant la face F_{Z+} on obtient (sans perte de généralité)

$$W_{F_{Z+}}(\vec{n}) = \iint_{[-1;1]^2} \left[\frac{(x \cdot \vec{n}_x + y \cdot \vec{n}_y + \vec{n}_z)}{(x^2 + y^2 + 1)^2} \text{HS}(x \cdot \vec{n}_x + y \cdot \vec{n}_y + \vec{n}_z) dx dy \right] \quad (1.3.2.1)$$

Le problème est alors de trouver un moyen de calculer, ou au moins d'approcher, la fonction $W_F(\vec{n})$. Ce calcul étant par ailleurs fait en chaque nœud du maillage, il est primordial de le faire en utilisant un minimum de ressource, quitte à évaluer une valeur approchée qui affectera le résultat de manière faible comparativement avec l'approximation faite précédemment et selon laquelle \mathcal{L} est constante sur chacune faces.

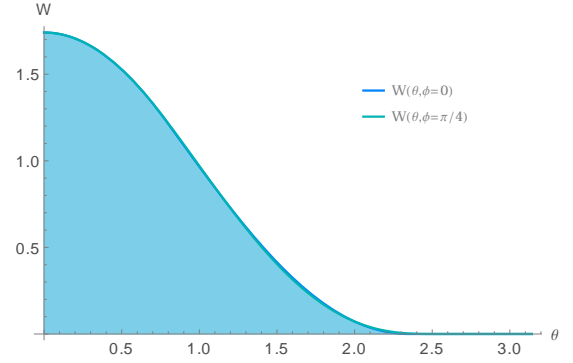
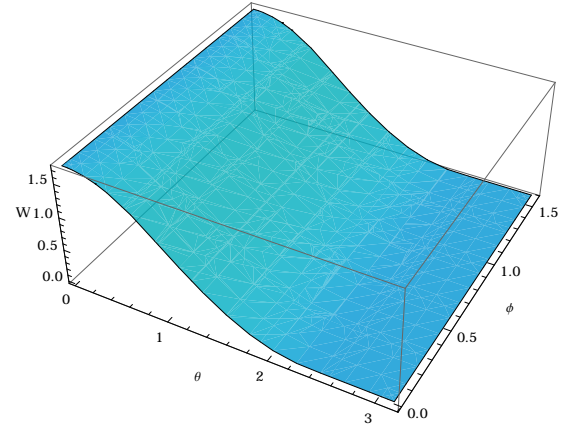

 (a) $W(\theta)$

 (b) $W(\theta, \phi)$

 FIGURE 1.3 – Évolution de $W_F(\vec{n})$ selon l'orientation de \vec{n}

Comme le montre la figure 1.3, la fonction W_F dépendant principalement de θ on tentera de l'approximer par une fonction de $\vec{n} \cdot \vec{F} = \cos(\theta)$

Une approximation simple est la fonction

$$\text{approx} : \cos(\theta) \mapsto \frac{[\max(.75 + \cos(\theta), 0)]^2}{1.75} \quad (1.3.2.2)$$

Comme le montre la figure 1.4, cette fonction est, malgré sa grande simplicité proche de la fonction $W_F(\vec{n})$.

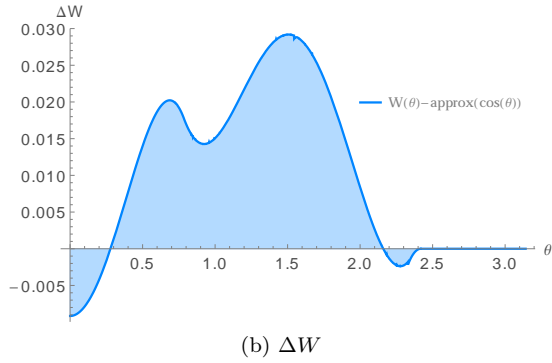
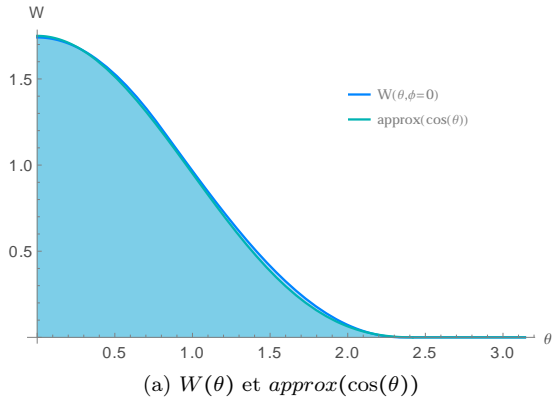


FIGURE 1.4 – Comparaison entre $W_F(\vec{n})$ et $\text{approx}(\cos(\theta))$

1.3.3

Discussion

La méthode développée ici repose sur l'approximation de la formule (1.3.1.10) et plus particulièrement de l'intégrale présenté dans la formule (1.3.1.8). La complexité de cette intégrale réside dans le domaine effectif d'intégration, conditionné à la fois par la forme carré des faces de l'envmap et la présence d'un terme caractéristique de l'hémisphère visible.

Plusieurs formules ont été étudiée pour des domaines d'intégration hémisphériques ou partiellement hémisphérique ainsi que pour des polygones entièrement visibles [?] (ce qui fait disparaître le terme de HS).

Il aurait été possible d'adapter un modèle polygonale évoqué dans [?] mais les polygones sur lesquels il aurait fallu intégrer variant selon les conditions de visibilité il aurait été nécessaire d'effectuer de coûteux calculs de géométrie.

L'objectif principal étant ici une grande vitesse d'exécution, l'inexactitude des résultats étant de toute façon largement oubliée au regard des approximations faites précédemment on préféra utiliser une fonction grossièrement approché mais simple à calculer.

Partie 1.4

Ombre

1.4.1

Modèle

Un indice visuel primordial à la vraisemblance visuel des images produites est la présence d'ombres portées provoquées par l'ajout de l'objet .

La modélisation de l'impact d'un tel ajout peut théoriquement être calculé qu'en connaissant la géométrie de la scène et la nature des matériaux qui la compose.

On fera ici plusieurs hypothèses dans le but d'obtenir un modèle qui soit calculable en temps réels tout en donnant des résultats vraisemblables.

Le calcul d'ombre douces est alors fait en décomposant l'objet en une hiérarchie de sphere comme présenté dans et en sommant les contribution des différentes sphères.

Pour évaluer l'ombre douce projeté par une sphere il suffit alors d'évaluer le manque d'illumination. On rappelle les formules suivantes

$$\mathcal{E}_{\text{ambient}}(p, \vec{n}) = \frac{1}{\pi} \int_{\mathcal{H}^2(\vec{n})} \mathcal{L}_{\text{glob}}(\vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega}$$

$$\mathcal{E}_{\text{ombre}}(p, \vec{n}) = \mathcal{E}_{\text{ambient}}(p, \vec{n}) - \frac{1}{\pi} \int_{\mathcal{S}(p)} \mathcal{L}_{\text{glob}}(\vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega}$$

avec $\mathcal{S}(p)$ la portion de sphere visible depuis la point considéré.

L'ombre est rendu en assombrissant le pixel associé ce point d'un facteur

$$\mathcal{F}(p) = \frac{\mathcal{E}_{\text{ombre}}(p, \vec{n})}{\mathcal{E}_{\text{ambient}}(p, \vec{n})} \quad (1.4.1.1)$$

$$= 1 - \frac{\int_{\mathcal{S}(p)} \mathcal{L}_{\text{glob}}(\vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega}}{\int_{\mathcal{H}^2(\vec{n})} \mathcal{L}_{\text{glob}}(\vec{\omega}) \times \vec{\omega} \cdot \vec{n} d\vec{\omega}} \quad (1.4.1.2)$$

Les différents niveaux de détail de l'envmap nous permettant d'obtenir des approximations de $L(p, \vec{\omega})$ sur $\mathcal{S}(p)$ et sur $\mathcal{H}^2(\vec{n})$ on peut simplifier la formule en :

$$\mathcal{F}(p) = 1 - \frac{\mathcal{L}_{\text{glob}}(\mathcal{S}(p))}{\mathcal{L}_{\text{glob}}(\mathcal{H}^2(\vec{n}))} \left(\frac{1}{\pi} \int_{\mathcal{S}(p)} \vec{\omega} \cdot \vec{n} d\vec{\omega} \right) \quad (1.4.1.3)$$

Le calcul de l'angle solide formé par la sphere, et pondéré par un cosinus est détaillé dans [?]. On retiendra que dans notre cas ou la sphere est supposé au dessus du plan sur lequel se projettent les ombres :

$$\int_{\mathcal{S}(p)} \vec{\omega} \cdot \vec{n} d\vec{\omega} = \cos(\omega) \sin^2(\alpha) \quad (1.4.1.4)$$

avec α le demi angle sous lequel est vu la sphere et ω l'angle entre la vertical (normale à la surface sur laquelle se projettent les ombres) et la direction de la sphere.

On obtient ainsi :

$$\mathcal{F}(p) = 1 - \cos(\omega) \sin^2(\alpha) \frac{\mathcal{L}_{\text{glob}}(\mathcal{S}(p))}{\mathcal{L}_{\text{glob}}(\mathcal{H}^2(\vec{n}))} \quad (1.4.1.5)$$

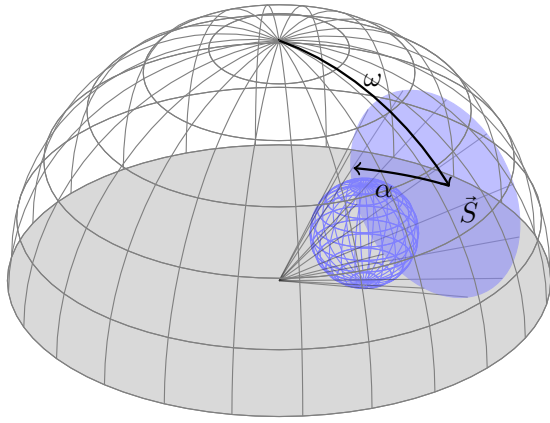


FIGURE 1.5 – Obstruction par une sphere

1.4.2

Discussion

La ou il est habituel de segmenter l'environnement pour en extraire une hiérarchie de sources lumineuses, la méthode proposée ici permet d'évaluer des ombres douces à partir de données d'environnement sans étape de segmentation ni calcul de visibilité.

Un développement intéressant serait de construire une décomposition hiérarchique, potentiellement intégrée dans un octree, et de l'évaluer plus ou moins profondément selon la distance considérée.

Partie 1.5

Reflets spéculaires

Le calcul de l'éclairage ambiant revient à considérer une BRDF purement lambertienne. Afin d'améliorer le réalisme du rendu il convient d'adopter un modèle de Phong en ajoutant une part d'éclairage spéculaire.

Cet éclairage spéculaire permet de rendre de manière intéressante des surfaces métallique, dans la limite d'un seul reflet. Le modèle adopté ici est par ailleurs isotrope, ce qui ne permet pas le rendu de matières comme du métal brossé pour lesquels on retrouve des directions privilégiés.

Dans notre cas l'évaluation du reflet se fait naturellement en calculant la réflexion du rayon incident, donné par la position du point considéré dans l'espace de la caméra, relativement à la normale de l'objet dans ce même espace. On accèdera ensuite à la valeur d'éclairage directement dans l'envmap.

Une fois de plus on pourra utiliser les différents niveaux de mipmap à notre avantage, la considération du niveau de mipmap revenant à considérer l'angle d'un cône autour de l'axe du reflet. Un tel cône permet ainsi de caractériser la spécularité de l'objet, cette dernière pouvant varier d'un reflet parfait –miroir– à un reflet plus diffus –plastique–. On utilisera également l'ombre projetée calculée précédemment afin de moduler les reflets.

Partie 1.6

Pipeline

Compte tenu des méthodes décrites précédemment, le pipeline de rendu se décompose en différentes parties (voir

figure 1.6)

1. L'étape de pré-calcul permet l'évaluation de données propres au modèle. Ces données n'étant pas influencées par la localisation dans l'espace ni par les caractéristiques d'environnement lumineux, il n'est pas nécessaire de les recalculer en temps réel et on préfère donc stocker les résultats pré-calculés.

Ambient : information d'auto-occultation ($\mathcal{P}_V(p)$), stocké dans une `texture2D` ;

Spheres : décomposition de l'objet comme union de sphères, stocké sous forme de `vec4[]`.

2. L'étape de calcul temps réel, qui évalue des résultats temporaires nécessaires à la réalisation du rendu final. Ces résultats doivent être réévalués dynamiquement car ils dépendent de paramètres dynamiques tels que les données d'environnement.

Ombre : ombre douce projetée par l'objet, elle dépend de l'environnement lumineux décrit par l'envmap.

3. L'étape de rendu qui produit l'image telle qu'elle est vue par l'utilisateur.

Rendu objet : affichage de l'objet, en tenant compte de l'éclairage ambiant, et des reflets spéculaires ;

Rendu ombre : affichage des ombres en surimpression afin d'intégrer l'objet de manière plus réaliste.

Partie 1.7

Résultats

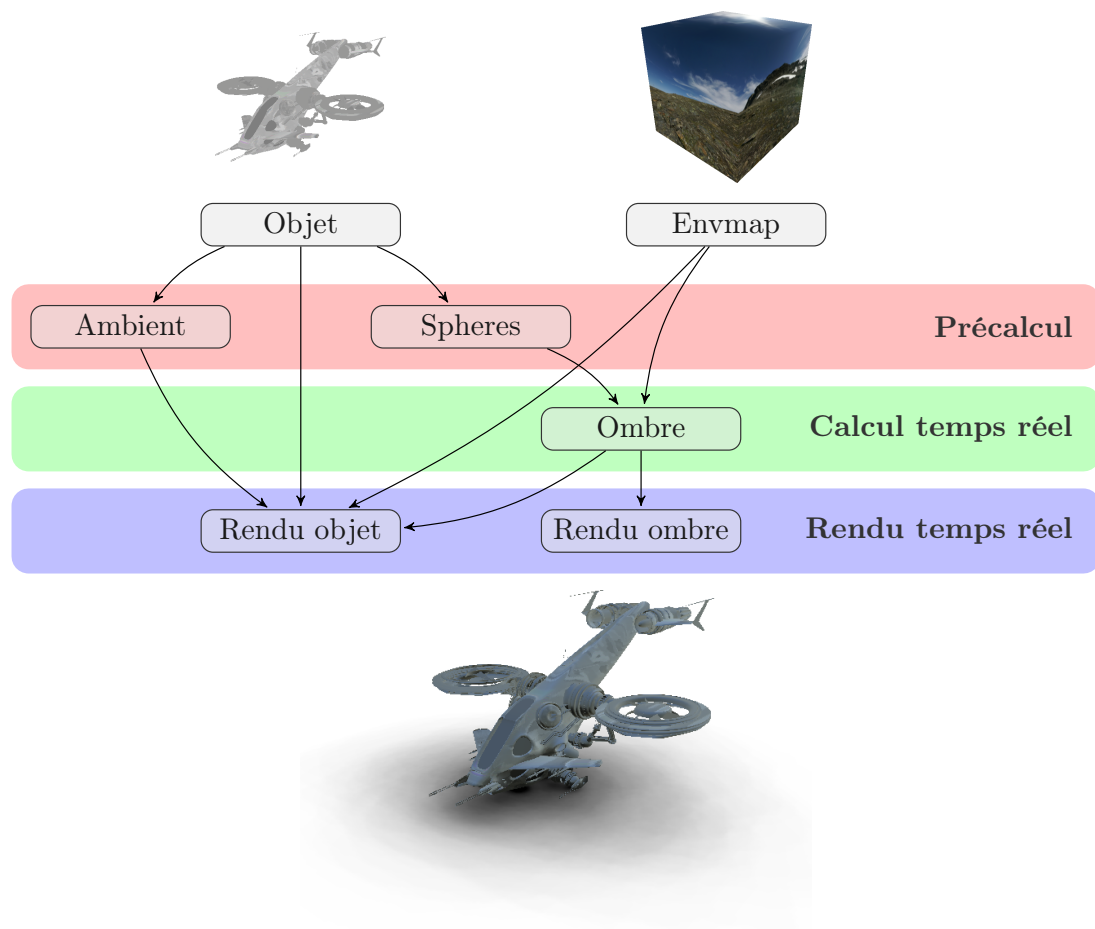


FIGURE 1.6 – Pipeline développé