HAL
open science

Éclairage direct en temps réel avec sources surfaciques et BRDF quelconques

Sylvain Meunier

To cite this version:

Sylvain Meunier. Éclairage direct en temps réel avec sources surfaciques et BRDF quelconques. Synthèse d'image et réalité virtuelle [cs.GR]. 2007. inria-00598444

HAL Id: inria-00598444

https://inria.hal.science/inria-00598444

Submitted on 6 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Institut National Polytechnique de Grenoble - Université Joseph Fourier M2R IVR 2007

Éclairage direct en temps réel avec sources surfaciques et BRDF quelconques

Sylvain MEUNIER

Rapport du projet de Master 2

Artis/Laboratoire LJK/IMAG-INRIA. UMR 5224.

Composition du jury :

Nicolas	HOLZSCHUCH	Directeur de stage
Eric	BRUNETON	Rapporteur externe
Augustin	LUX	Membre du jury
Pierre	BESSIÈRE	Membre du jury

Table des matières

1 État de l'art **5**
1.1 Rendu photo-réaliste 5
1.1.1 Matières 5
1.1.2 Représenter une matière opaque 6
1.1.3 Modèles de BRDF 8
1.1.4 Équation du rendu 10
1.1.5 Simplifications courantes de l'équation du rendu 12
1.2 Techniques de rendu 13
1.2.1 Précalculs, *Lightmaps* et fonction plénoptique 13
1.2.2 Factorisation de BRDF 13
1.2.3 Précalculs et *environment maps* 14
1.2.4 Précalculs et harmoniques sphériques 17
1.2.5 Précalculs et ondelettes 21
1.3 Bilan 22

2 Réalisation **24**
2.1 Cadre du stage 24
2.2 Approches caduques 25
2.2.1 Précalculer l'équation du rendu 25
2.2.2 Harmoniques sphériques 25
2.3 Approche brutale 26
2.4 Amélioration de la première approche 29
2.4.1 Visualiser le problème 29
2.4.2 Déterminer l'échantillonnage nécessaire 30
2.4.3 Comparaison entre la méthode brutale et la méthode adaptative 38
2.5 Approche biaisée 41
2.6 Approche probabiliste 45
2.6.1 Intégration probabiliste 45
2.6.2 Monte Carlo et équation du rendu 47
2.7 Notre intégration probabiliste 48

2.7.1 Précalculer P 48
2.7.2 Une méthode de rejet 48
2.7.3 Éviter ces calculs inutiles 49
2.7.4 Discussion 50

3 Conclusion **54**
3.1 Bilan et Perspectives 54
3.2 Remerciements 56

Introduction

L'augmentation rapide de la flexibilité et des performances offertes par la nouvelle génération de cartes graphiques programmables permet d'envisager d'inclure dans des applications temps-réel des effets autrefois limités à des simulations d'éclairage caractérisées par de longs temps d'exécution. Parmi ces effets, on retrouve les ombres douces, des réflections réalistes ou l'anisotropie des matières comme l'aliminium brossé.

Lorsque l'on travaille dans le domaine de la synthèse d'images réaliste et interactive, on recherche toujours le meilleur compromis entre qualité et rapidité. Il est donc nécessaire de réduire les calculs au strict minimum tout en conservant du réalisme. Toute la difficulté réside dans le fait que chaque effet nécessite des simplications et des optimisations spécifiques.

Les sources surfaciques proches sont rarement considérées pourtant elles sont présentent dans de nombreuses scènes. Les reflets causés par une lumière de ce type sont particulièrement intéressants car ils prennent sa forme. Considérer ces sources lumineuses apporte plus de réalisme mais impose de nouvelles limitations et ce rapport propose une solution simple et efficasse pour y parvenir.

Plan Dans le premier chapitre, nous étudierons un large panel de travaux existants et nous motiverons le sujet de ce stage. Puis dans le chapitre suivant, nous décrirons et discuterons le travail réalisé jusqu'ici. Enfin, nous tâcherons d'établir un bilan et donnerons quelques perspectives à ce travail.

Chapitre 1

État de l'art

Dans une première partie, nous rappellerons brièvement les bases du rendu photo-réaliste en éclairage. Puis nous décrirons un large éventail de méthodes s'en inspirant dans l'optique d'application temps-réel. Enfin, nous situerons le cadre du stage effectué.

1.1 Rendu photo-réaliste

Cette partie introduit l'équation du rendu proposée par Kajiya [10]. Elle est la base de toutes les approches présentées. Nous décrirons rapidement les types de matières existants. Puis nous nous intéresserons à la description des matières opaques en présentant quelques modèles de réflectance pratiques. Enfin, nous définirons l'équation du rendu et verrons quelques simplifications courantes.

1.1.1 Matières

Les matières des objets jouent un rôle fondamental dans notre perception du monde. Dans cette partie, nous décrirons rapidement les différents types de matières existants.

Matières opaques

Les matières opaques empêchent la lumière de pénétrer dans l'objet et se contentent de la réfléchir ou de l'absorber à la surface. Elles sont séparables en deux groupes : les matières opaques hétérogènes (telles que le bois) dont les propriétés optiques changent sur la surface et les matières opaques homogènes (la craie).

On peut distinguer trois types de réflections : les réflections diffuses dont le résultat ne dépend pas du point de vue ; les réflections spéculaires pour lesquelles

la lumière est réfléchie dans une seule direction (comportement similaire à un miroir) ; et les réflections brillantes (*glossy*) qui représentent tous les matériaux entre diffus et spéculaires. Tous les matériaux réels sont à ranger dans cet catégorie, les modèles diffus et spéculaire étant purement théoriques et absents dans la nature.

Il existe deux types de matières opaques : les matières diélectriques et les matières métalliques. Le reflet d'un diélectrique a la couleur de la source lumineuse alors que la couleur du reflet d'un métal est influencée par celui ci. De plus les diélectriques réfléchissent plus de lumière lorsque le point de vue est rasant alors que les métaux réfléchissent à peu près toujours la même quantité lumineuse. Cet effet est décrit par la loi de Fresnel. Elle explicite la répartition du flux lumineux entre réflection, réfraction et absorption lorsque celui ci entre en contact avec l'objet.

Matières translucides

Contrairement aux matières opaques, les matières translucides laissent entrer une partie de la lumière. Elle entre dans la matière, est dispersée et ressort. Les photons peuvent entrer et sortir n'importe où. La peau en est un très bon exemple. Si on place sa main près d'une lampe, on verra les bords de sa main et de ses doigts devenir rouge.

Matières transparentes

Les matières transparentes sont un cas particulier des matières translucides. La différence existant entre ces deux types de matières est similaire à celle existant entre une matière brillante et une matière spéculaire. Une matière transparente ne disperse pas la lumières qui la traverse.

Dans la suite de ce rapport nous traiterons exclusivement de matières opaques.

1.1.2 Représenter une matière opaque

Pour représenter une matière opaque, on lui associe une fonction de réflectance appelée BRDF (*Bidirectional Reflectance Distribution Function*). La quantité décrivant le flux lumineux arrivant ou quittant un élément de surface par un angle solide est appelée radiance. La fonction de réflectance est définie comme étant le rapport de la radiance L_o quittant la position x dans la direction ω_o par l'irradiance E en x par angle solide :

$$E(x, \omega_i) = \int_{\Omega+} L_i(x, \omega_i) cos(\theta_i) d\omega_i \tag{1.1}$$

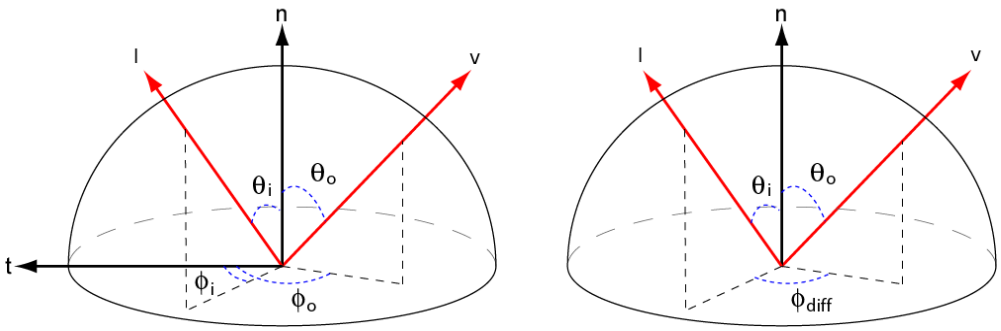

FIG. 1.1 – Cette figure illustre les différents paramètres d'une BRDF. t est le vecteur tangent. Ces images sont extraites d'une présentation de Addy Ngan.

$$f_r(x,\omega_i,\omega_o) = \frac{dL_o(x,\omega_o)}{dE(x,\omega_i)} = \frac{dL_o(x,\omega_o)}{L_i(x,\omega_i)cos(\theta_i)d\omega_i} \tag{1.2}$$

ω_o et ω_i sont des directions sur la sphère unité et x est une position 2D sur la surface. ω_o est la direction de l'oeil de l'utilisateur. ω_i est la direction de la lumière incidente. Il est aussi possible d'exprimer les directions ω_o et ω_i sous la forme de vecteurs normés. L_i représante la radiance incidente. La BRDF peut varier en fonction de la longueur d'onde, mais pour des raisons pratiques on se contente généralement de la définir pour les trois canaux rouge, vert et bleu.

Lorsque la BRDF dépend de tous ses paramètres, elle est appelée SBRDF (*Spatially varying Bidirectional Reflectance Distribution Function*) et permet de représenter les matières opaques hétérogènes.

Une matière est dite anisotropique si sa BRDF change quand la surface tourne autour de sa normale (un objet en aluminium brossé par exemple), dans le cas contraire elle est dite isotopique. La paramétrisation d'une BRDF isotropique est paramétrisable comme ceci pour gagner une dimension :

$$f_r(\theta_i,\theta_0,\phi_i-\phi_o) = f_r(\theta_i,\theta_0,\phi_{diff}) \tag{1.3}$$

En prenant $\omega_i = (\theta_i,\phi_i)$ et $\omega_o = (\theta_o,\phi_o)$.

L'utilisation d'une BRDF impose quelques restrictions. Certains effets comme la phosphorescence (réémission non instantanée des photons) ou la fluorescence (modification du spectre lors de la réflection) ne sont pas représentés. De plus elle doit respecter trois règles pour être physiquement plausible : être positive, respecter la réciprocité de Helmholtz et la conservation de l'énergie [18].

1.1.3 Modèles de BRDF

Il existe de nombreux modèles analytiques permettant de représenter une BRDF. Certains modèles sont inspirés de lois physiques alors que d'autres prennent enormément plus de libertés. La plupart du temps les paramètres de ces modèles sont retrouvés à partir de données mesurées. Un bon comparatif de l'efficacité des principaux modèles est réalisé par Ngan *et al.* [23]. Ici, nous ne décrirons donc que les modèles les plus couramment utilisés sur GPU.

Modèle de Lambert

$$f_r = \frac{\rho_d}{\pi} \tag{1.4}$$

Ce modèle représente la BRDF avec un seul paramètre : un paramètre ρ_d représentant la réflectance de l'objet. π est un facteur de normalisation. Il permet de représenter les matières opaques strictement diffuses et sert de base à de nombreux modèles comme nous allons le voir.

Modèle de Phong

$$f_r(l, v) = \frac{\rho_d}{\pi} + \frac{\rho_s}{n \cdot l}(r \cdot l)^e \tag{1.5}$$

v est le vecteur en direction sortante (oeil de l'utilisateur par exemple). l est le vecteur en direction de la lumière incidente. r est le vecteur réfléchi de v par rapport à la normale n en x.

Ce modèle [24] possède deux parties distinctes et représente la BRDF avec 3 paramètres : un paramètre constant ρ_d pour la partie diffuse, un paramètre constant ρ_s et un exposant e pour la partie spéculaire. Une matière légèrement brillante aura un exposant faible et de grands reflets, alors qu'une matière spéculaire aura un exposant élevé et de petits reflets. La partie spéculaire de la fonction a l'apparence d'un lobe. Ce modèle donne un aspect métallique aux objets et ne permet pas de capturer un effet comme la rétro-réflection, ni de représenter une matière anisotropique et n'est pas physiquement exact du tout.

Modèle de Blinn-Phong

$$f_r(l, v) = \frac{\rho_d}{\pi} + \frac{\rho_s}{n \cdot l}(h \cdot n)^e \tag{1.6}$$

h est le vecteur milieu. Il est défini comme ceci :

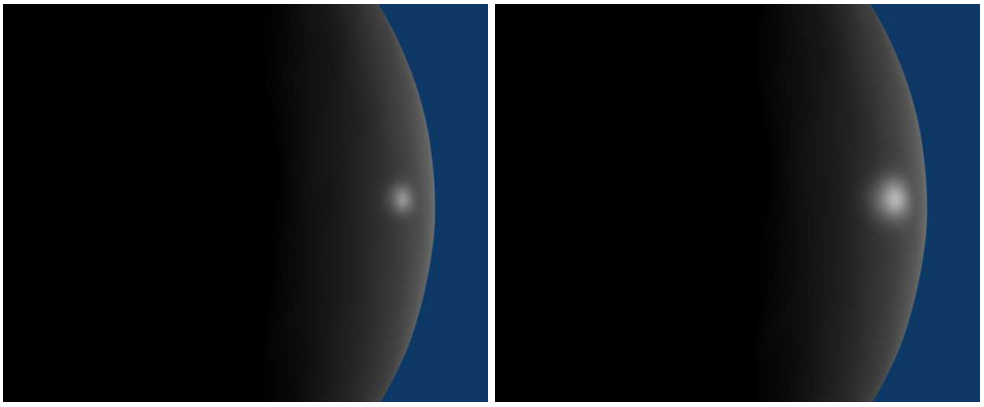

Fig. 1.2 – Cette figure permet de comparer les reflets obtenus avec le modèle de Phong (à gauche) et celui de Blinn-Phong (à droite) pour un exposant $e = 200$. Les deux reflets sont nettement différents et on peut voir l'augmentation de la spécularité du modèle de Blinn (reflet plus petit).

$$h = \frac{v + l}{||v + l||} \tag{1.7}$$

La principale différence du modèle de Blinn-Phong [3] avec le modèle de Phong est l'utilisation du vecteur milieu h à la place du vecteur réfléchi r. Cette modification réduit le coût du calcul et modifie le comportement des reflets aux angles rasants. Le lobe du modèle de Blinn-Phong rétrécie pour ce genre de point de vue. Ce comportement donne l'impression que la spécularité augmente et est proche des matières diélectriques comme le plastique (voir figure 1.2).

Modèle de Lafortune

$$f_r(l^l, v^l) = \frac{\rho_d}{\pi} + \sum_i (C_{x,i} l^l_x v^l_x + C_{y,i} l^l_y v^l_y + C_{z,i} l^l_z v^l_z)^{e_i} \tag{1.8}$$

Le suffixe ^{l} indique que les vecteurs sont placés dans un repère local de la surface.

Ce modèle [17] est une généralisation du modèle de Phong. Il permet d'approcher efficacement une BRDF mesurée par une somme de plusieurs lobes et d'une partie diffuse. Chaque lobe est défini par trois coefficients de forme C_x, C_y et C_z qui permettent d'orienter le lobe, et un exposant e. Si on choisi les coefficients de forme tel que $C_x = C_y = -1$ et $C_z = 1$ on retrouve le modèle de Phong à un facteur de normalisation prés. Si on utilise $C_x = C_y > 0$ on obtient une rétro-réflection. En

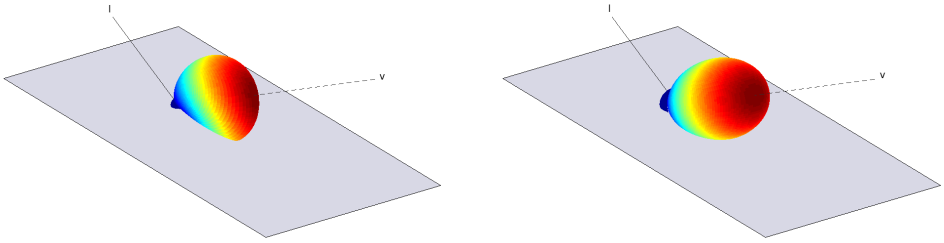

FIG. 1.3 – Cette figure permet de comparer la forme des lobes des modèles de Blinn-Phong et de Phong pour le même exposant avec un point de vue rasant. À gauche, le lobe de Blinn-Phong. À droite, le lobe de Phong. Ces images sont extraites d'une présentation de Addy Ngan.

attribuant des valeurs différentes à C_x et C_y différemment, on obtient des matières anisotropiques.

La figure 1.3 montre la différence entre un lobe du modèle de Blinn-Phong et un lobe du modèle de Phong (ou de Lafortune).

1.1.4 Équation du rendu

L'illumination d'une scène composée d'objets ayant des matières opaques est habituellement formulée par l'équation du rendu. Une définition précise (rappels de radiométrie, unités...) est présente dans [18]. Voici la première formulation exprimée par rapport aux surfaces :

$$L_o(x,v) = L_e(x,v) + \int_S f_r(x,l^l,v^l)G(x,y)V(x,y)L_o(y,-l)dA_y \tag{1.9}$$

Il s'agit d'une intégrale sur toutes les surfaces S de la scène. Le suffixe ^{l} précise que le vecteur ou le point est représenté dans un repère local. Des vecteurs ou des points sans suffixe indiquent que l'on peut tous les exprimer dans un repère quelconque. Les vecteurs sont normés. $L_o(x,v)$ est la radiance perçue au point x par la direction v. $L_e(x,v)$ est la radiance émise au point x dans la direction v. $f_r(x,l^l,v^l)$ est la BRDF de la surface considérée. $V(x,y)$ est une fonction prédisant la visibilité du point y depuis x. $L_o(y,l)$ est la radiance émise en y en direction de x. $G(x,y)$ est un facteur géométrique permettant de faire le lien entre les angles solides et les éléments de surface :

$$G(x,y) = \frac{cos(\theta_x)cos(\theta_y)}{||x-y||^2} \tag{1.10}$$

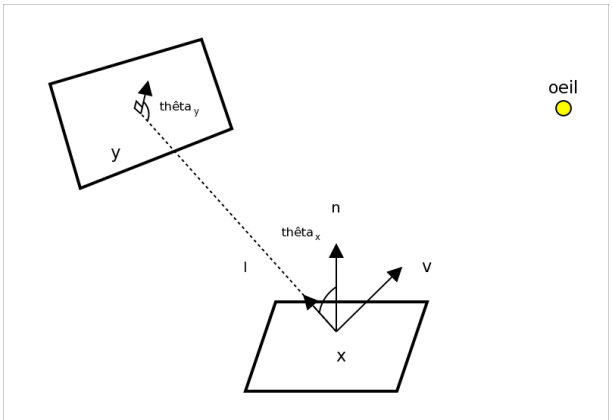

FIG. 1.4 – Illustration de l'équation du rendu.

θ_x est l'angle entre la normale de la surface au point x et le vecteur $l = \overrightarrow{xy}/\|xy\|$. θ_y est l'angle entre la normale en y et la direction $-l$ (voir figure 1.4).

Connaissant la radiance émise en chaque point des surfaces de la scène, calculer l'illumination d'une scène revient à chercher la radiance perçue en chaque point pour obtenir l'équilibre de l'équation 1.9. Ce qui est très complexe.

En discrétisant cette équation, on peut l'écrire sous une forme plus compacte :

$$L_o = L_e + \Gamma L_o \tag{1.11}$$

Γ est l'opérateur de transport de la lumière.

Une solution formelle est obtenue par les séries de Neumann :

$$L_o = \sum_{i=0}^{\infty} \Gamma^i L_e \tag{1.12}$$

Cette série est interprétable comme la somme de la lumière émise, de la lumière réflectée une fois, de la lumière réflectée deux fois...

Voici une deuxième formulation cette fois-ci axée sur la radiance incidente :

$$L_o(x,v) = L_e(x,v) + \int_{\Omega+} f_r(x,l^l,v^l) L_i(x,l) cos(\theta_x) dl \tag{1.13}$$

$\Omega+$ est l'hémisphère supérieure au point x. L_i est la radiance incidente.

Ces deux formulations sont utiles dans des cas différents. Pour travailler sur un algorithme de radiosité ou sur de l'éclairage direct, on aura tendance à utiliser l'équation 1.9. Alors que dans le cas des méthodes temps réel à base d'*environment maps* que nous décrirons dans la suite de cet état de l'art, l'équation 1.13 sera plus adaptée.

La quantité observée en chaque pixel au moment du rendu, est l'irradiance du pixel intégrée sur son cône de vision. Une simplification très courante est d'observer la radiance réflectée par la surface au point touché par le rayon passant par le centre de ce pixel. Pour obtenir la couleur, il suffit de calculer la radiance pour chaque canaux de couleur, puis de prendre en compte le temps d'exposition (une simple multiplication par la valeur choisie).

1.1.5 Simplifications courantes de l'équation du rendu

L'équation du rendu est souvent simplifiée par quelques hypothèses bien choisies. Nous allons en détailler quelques unes.

Le terme ambiant

L'illumination indirecte est souvent approximée par l'ajout d'un terme ambiant. L'équation du rendu 1.13 devient :

$$L_o(x,v) = k_a L_a + L_e(x,v) + \int_{\Omega+} f_r(x,l^l,v^l) L_{direct}(y,-l) cos(\theta_x) dl \tag{1.14}$$

L_a est la radiance moyenne émise dans la scène. k_a est une constante. La radiance incidente L_{direct} ne provient que des sources de lumière visibles. Comparé à un calcul de l'éclairage indirecte, utiliser cette approximation est gratuite.

Éclairage distant

En faisant l'hypothèse que la radiance incidente provient de sources de lumière placées à l'infini. L_i ne dépend que de la normale sur la surface. L'équation du rendu 1.13 devient :

$$L_o(x,v) = L_e(x,v) + \int_{\Omega+} f_r(x,l^l,v^l) L_{infini}(l) cos(\theta_x) dl \tag{1.15}$$

L_{infini} est facilement représentable par une *environment map*. Cette simplification réduit la complexté de l'intégrale et elle est la base de beaucoup des méthodes suivantes.

Lumière ponctuelles

Éclairer la scène uniquement avec des lumières ponctuelles réduit énormément les temps de calcul. En effet la coûteuse intégration idéalement continue est remplacée par une somme discrète dont le nombre de termes est directement lié au

nombre de sources lumineuses. En combinant cette simplification avec un terme ambiant et une matière non émissive, on obtient :

$$L_o(x,v) = k_a L_a + \sum_i f_r(x, l^l, v^l) \frac{I_i}{||x - p_i||^2} V(x, p_i) cos(\theta_x) \tag{1.16}$$

p_i est la position de $i$ème la lumière ponctuelle et I_i son intensité. Les cartes graphiques actuelles intègrent un calcul proche de celui ci.

Après ces quelques rappels, nous décrirons un large éventail (non exhaustif) des méthodes interactives utilisant des BRDF.

1.2 Techniques de rendu

Il existe de nombreuses approches pour faire du rendu interactif ou temps-réel. Ces méthodes cherchent toutes à résoudre l'équation du rendu avec le meilleur compromis entre rapidité et qualité.

1.2.1 Précalculs, *Lightmaps* et fonction plénoptique

L'utilisation de *lightmaps* [6] est certainement l'approche la plus ancienne. Elle consiste à réaliser complètement la simulation d'éclairage et d'en stocker le résultat sous la forme de textures. Cette approche impose que les matières soient diffuses pour que le résultat ne dépende pas du point de vue. Une fois calculées, le rendu est un simple placage de textures. Lorsque les matières ne sont pas simplement diffuses, on parle alors de fonction plénoptique [8]. Cette fonction encode toutes les images possibles de la scène (4 dimensions, 2 paramètres pour la position du point observé, 2 pour le point de vue utilisé, lorsque l'on travaille en RGB). Le problème majeur de ces approches est l'utilisation de scènes totalement statiques, les objets mobiles ne peuvent pas bénéficier de l'éclairage de la scène. De plus la représentation d'effets très localisés ou la visualisation de grandes scènes demande beaucoup de mémoire (textures en haute résolution).

1.2.2 Factorisation de BRDF

Les méthodes suivantes simplifient le problème en ne considérant qu'un éclairage direct par des sources ponctuelles ou directionnelles. L'utilisation directe de BRDF mesurées peut être réalisée avec une texture à 4 dimensions (2 angles par direction), mais le coût mémoire est très élevé pour un faible échantillonnage (environ 50MO pour 64^4 échantillons de 3 octets chacun). De plus elle ne permet

pas une implémentation sur GPU. Néanmoins plusieurs méthodes à base de factorisation ont été proposées pour compresser ces données et faciliter leur utilisation.

Kautz et McCool [12] utilise une factorisation de la BRDF pour réaliser un rendu interactif avec n'importe quel type de BRDF. Une fois la BRDF factorisée, les facteurs obtenus (deux fonctions $\mathbb{R}^2 \to \mathbb{R}$) sont représentés par deux textures à deux dimensions. Ces textures sont paramétrées comme les *parabolic maps* de Heidrich et Seidel [9]. Le rendu est réalisé en utilisant le placage de textures et le *blending* standard offert par OpenGL. Prenons le cas où il n'y a qu'une lumière. Le rendu est un simple produit par pixel des facteurs et d'un terme en cosinus. L'évaluation d'un facteur est un accès texture, le produit est fait par un *blending* en mode multiplicatif avec plusieurs passes et le terme en cosinus est calculé avec l'activation d'une matière diffuse.

Kautz montre que l'utilisation de la décomposition en valeurs singulières pour factoriser numériquement la BRDF est très coûteuse sans être adaptée au rendu. En effet son utilisation fait apparaître des valeurs négatives sur la BRDF reconstruite. De plus elle est inadaptée aux anciennes cartes graphiques (G70 et plus vieux) car les facteurs peuvent être négatifs. Il propose donc d'utiliser la décomposition normalisée. Cette décomposition, moins coûteuse, assure que la BRDF reconstruite soit positive et que les facteurs obtenus soient positifs. La factorisation est effectuée par plusieurs méthodes avec différentes paramétrisations des BRDF. Kautz remarque aussi que la paramétrisation utilisée lors de la décomposition affecte la qualité du résultat et montre qu'une version modifiée de la paramétrisation de Rusinkiewicz [26] est très efficace.

La qualité et l'efficacité des décompositions précédemment citées ne sont pas satisfaisantes. En effet, les BRDF reconstruites ne respectent pas forcement les principes physiques de base. McCool *et al.* [20] propose un nouvel algorithme de décomposition pour palier à ce problème et ainsi obtenir une factorisation de meilleure qualité. McCool choisit d'approximer la BRDF comme un produit de plusieurs facteurs, alors que Kautz ce limite au produit de deux facteurs. Une fois la BRDF factorisée, la méthode de rendu utilisée est similaire à celle donnée par Kautz.

On remarquera que la technique de rendu en plusieurs passes est remplaçable par une technique en une passe en utilisant un *fragment program*.

1.2.3 Précalculs et *environment maps*

L'extension de ces méthodes aux sources de lumière surfaciques était inefficace à cette époque. Il a donc été proposé de précalculer et de tabuler l'équation du rendu. Ces méthodes font toutes l'hypothèse que les sources de lumière se trouvent infiniment loin et utilisent des *environment maps* [4] pour stocker la radiance réflectée. Ces *environment maps* contiennent une version préfiltrée de la ra-

diance incidente par rapport à la BRDF, elle-même stockée initialement avec une *environment map*. Le choix du préfiltrage se fait toujours de la même manière. Les auteurs cherchent à réduire le coût mémoire, en effet l'équation de rendu dépend de 5 paramètres dans le cas général. Dans un premier temps, ils se limitent à un type de BRDF intéressant (modèle de Phong...). Utilisant cette restriction, ils simplifient l'équation du rendu 1.13 en supprimant les termes devenus inutiles...Puis ils stockent le résultat de l'équation simplifiée. Enfin lors du rendu, le coûteux calcul de l'équation de rendu est remplacé par une ou plusieurs lectures dans des textures. Nous allons décrire quelques techniques.

Modèle de Phong

Miller [7], Heidrich et Seidel[9] proposent deux solutions similaires permettant l'utilisation du modèle de Phong. La partie diffuse et la partie spéculaire de la BRDF sont stockées avec deux *environment maps*, puis lors du rendu, les lectures sont combinées grâce à un terme de Fresnel. Voici la simplification de l'équation 1.13 pour la partie diffuse :

$$L_o(x,v) = \int_{\Omega+} f_r(x,l^l,v^l)L_{infini}(l)cos(\theta_x)dl = \frac{\rho_d}{\pi}\int_{\Omega+} L_{infini}(l)cos(\theta_x)dl \quad (1.17)$$

On remarque que le résultat ne dépend que de la normale à la surface (via le terme $cos(\theta_x)$), ce qui permet de précalculer l'équation pour "toutes" les normales possibles. Le résultat est stocké dans une *environment map* à deux dimensions (deux coordonnées suffisent pour représenter un vecteur normé).

Pour la partie spéculaire du modèle de Phong, on part de la même équation 1.13, on substitue f_r et on remarque que le résultat ne dépend que du vecteur réfléchi...

Cette méthode permet un rendu temps réel avec des machines peu puissantes, mais les matières représentables sont extrêmement limitées par l'utilisation du modèle de Phong.

BRDF isotropiques 1

Kautz et McCool [13] étendent l'idée précédente en considérant des BRDF isotropiques symétriques par rapport à l'axe de réflection. Utilisant le fait que la BRDF traitée est clairement spéculaire (la BRDF est non nulle seulement quand $r \approx l$), ils sortent le terme en cosinus de l'intégrale et obtiennent :

$$L_o(x,v) = \int_{\Omega+} f_r(x,l^l,v^l)L_{infini}(l)cos(\theta_x)dl = (n \cdot r)\int_{\Omega+} f_r(x,l^l,v^l)L_{infini}(l)dl \quad (1.18)$$

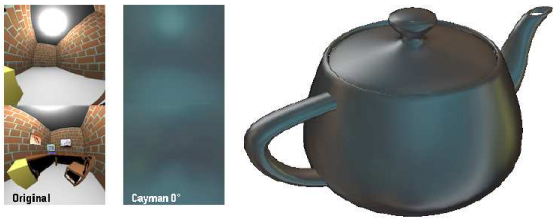

FIG. 1.5 – Cette figure illustre la méthode de Kautz et McCool. On peut voir l'*environment map*, la version filtrée et le résultat sur une theillère.

Utilisant le fait que le comportement de la BRDF suit la loi de Fresnel (la matière réfléchit plus la lumière aux angles rasants), ils proposent cette approximation de la BRDF :

$$f_r(l,v) \approx F(n,r)p(r,l) \tag{1.19}$$

F est le terme de Fresnel. p est une fonction définissant un lobe constant.

Grâce à cette approximation, ils arrivent à la simplification suivante de l'équation de rendu :

$$L_o(x,v) = (n \cdot r)F(n,r) \int_{\Omega+} p(r,l)L_{infini}(l)dl \tag{1.20}$$

Une *environment map* à deux dimensions (pour v) stocke le précalcul de l'intégrale. Lors du rendu on calcule le terme de Fresnel et le produit scalaire, puis on les multiplient avec la valeur lue dans l'*environment map*. D'une manière similaire à Miller et Heidrich, les BRDF plus complexes composées de plusieurs phénomènes séparables (rétro-réflection, réflection diffuse...) sont utilisables avec un rendu en plusieurs passes en décomposant la BRDF en plusieurs BRDF simples. Leur méthode ne perd aucun avantage des méthodes proposées par Miller et Heidrich, mais permet l'utilisation d'une plus large palette de matières.

McAllister *et al.* [19] propose une solution similaire avec des SBRDF en utilisant le modèle de Lafortune [17]. Il précalcule une douzaine d'*environment maps* pour gérer toutes les BRDF de la surface. La simplification de l'équation est réalisée en considérant l'axe du maximum de chaque lobe.

BRDF isotropiques 2

Partant du même type de BRDF, Cabral *et al.* [5] préfiltre une *environment map* pour plusieurs angles de vue. Il obtient plusieurs *environment maps* à 4 dimensions (2 pour le point de vue, 2 pour la normale) dont peu d'angles de vue ont été stockés. Les angles de vue non stockés sont générés avec une méthode de *warping*

lors du rendu. La solution proposée par Cabral est nettement moins efficace que celle de Kautz et McCool. En effet, l'opération de déformation n'est réalisable que sur une machine très performante pour espérer avoir un rendu interactif. De plus, la projection de la géométrie sur l'écran doit être orthogonale car Cabral fait l'hypothèse que le spectateur est à l'infini pour réaliser son *warping*.

Préfiltrer des *environment maps*

Toutes les méthodes précédentes nécessitent un préfiltrage. Kautz *et al.* [15] proposent une méthode hiérarchique permettant de réduire considérablement le temps de précalcul. Ils effectuent le filtrage sur une pyramide de niveau de détails de l'*environment map*. Remarquant que le filtre de convolution est généralement lisse, ils en déduisent qu'il n'est pas nécessaire d'effectuer le filtrage au meilleur niveau de détail pour toute l'*environment map* et réalisent donc un filtrage adaptatif. Ils décrivent aussi comment effectuer un préfiltrage interactif avec OpenGL pour le modèle de Phong. Leur méthode n'est efficace que sur des matières très spéculaires (petit noyau de convolution).

BRDF anysotropiques

Dans [15], Kautz *et al.* apportent une dernière contribution, ils utilisent le modèle de Banks pour représenter des BRDF anisotropiques et n'utilisent qu'une *environment map* à 3 dimensions (2 pour le vecteur tangent et 1 pour un scalaire dépendant de la tangente et du point de vue) pour effectuer le rendu.

Conclusion Ces différents articles font apparaître la principale limitation de ce genre d'approche. Elles nécessitent de bonnes capacités de convolution et les cartes graphiques actuelles ne le proposent pas. De plus il est impossible d'inclure des effets comme l'ombrage ou les inter-réflections (réflections lumineuses de l'objet sur lui même).

1.2.4 Précalculs et harmoniques sphériques

Kautz *et al.* [14] propose une approche utilisant une base d'harmoniques sphériques pour résoudre l'équation du rendu.

Rendu et harmoniques sphériques

Les harmoniques sphériques [14][30] permettent de représenter des fonctions sur la sphère, comme les BRDF ou une *environment map*, dans un domaine fréquentiel. Elles sont un équivalent des fonctions sinus et cosinus utilisées par la

transformé de Fourier. De plus elles possèdent deux propriétés très intéressantes : leurs utilisation pour représenter l'éclairage et les BRDF réduit l'équation du rendu à un produit scalaire [5][22] (base orthonormée) ; il est possible d'effectuer la rotation d'une fonction sans avoir à la reprojeter avec une application linéaire (via une multiplication avec une matrice). Dans le cas d'un environnement composé de faibles fréquences (sources lumineuses étendues, matières brillantes ou diffuses...), Kautz montre que l'utilisation de 25 coefficients (5 bandes d'harmoniques sphériques) pour représenter la BRDF et la source lumineuse est suffisante et permet un rendu interactif. Les différentes méthodes utilisant les harmoniques sphériques n'adressent que les environnement de basses fréquences.

Les BRDF sont représentées par une table de coefficients. Cette table permet de stocker les coefficient pour chaque angle de vue ω_o.

Une fois la BRDF et la radiance incidente (une *environnement map*, une lumière circulaire ...) projetées dans la base des harmoniques sphériques, l'algorithme de rendu revient à : effectuer un changement de repère (une rotation) des coefficients de la radiance incidente vers le repère local ; effectuer un changement de repère (une rotation) du vecteur de vue vers le repère local ; récupérer les coefficients de la BRDF pour le point de vue courant ; calculer le produit scalaire des coefficients de la radiance incidente avec ceux de la BRDF.

La rotation des coefficients de la radiance incidente n'est pas réalisable sur GPU car le produit matriciel est rédhibitoire, Kautz réalise ce calcul sur CPU puis transmet les coefficients transformés au GPU pour qu'il puisse achever le rendu. Cette étape est le goulot d'étranglement de la méthode. On peut néanmoins accélérer le rendu en fixant le point de vue ou la lumière, ce qui permet de précalculer les changements de repère.

La méthode présentée par Kautz permet d'effectuer le rendu interactif d'une scène dynamique avec de larges sources de lumières et des BRDF quelconques.

Il y a néanmoins des problèmes liés à l'utilisation des harmoniques sphériques : le changement de repère de coefficients impose des scènes statiques pour permettre une déambulation temps réel ; les hautes fréquences ne sont pas utilisables (BRDF très spéculaires et/ou éclairages nets) ; de plus des artéfacts visuels (oscillations appelées phénomène de Gibbs, voir figure 1.6) apparaissent quand les fréquences représentant une grande partie de l'énergie de l'éclairage sont "presque représentables" par le nombre de coefficients utilisé (cas où les fréquences intéressantes sont représentées par les derniers coefficients).

Transfert de radiance précalculé et harmoniques sphériques

Au même moment, Sloan *et al.* [30] propose une approche à base d'harmoniques sphériques permettant de capturer des effets d'illumination globale comme l'auto-ombrage, les inter-réflections et les caustiques provoquées par l'objet sur

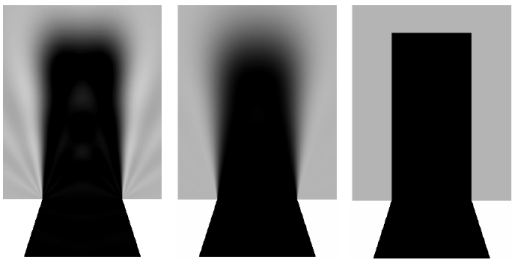

FIG. 1.6 – Cette figure illustre le phènomêne de Gibbs. À gauche, l'image de référence. À droite, l'image contenant les ocsillations. Au centre, le résultat obtenu après avoir réalisé un filtre passe bas. Ces images sont extraites de d'un article de Peter-Pike Sloan. Ces images sont extraites d'un article de Sloan.

lui même ou sur son voisinage. Cette méthode s'intègre parfaitement à la solution proposée par Kautz.

Ces effets sont précalculés par un algorithme d'illumination globale, puis la opérateur de transport est stocké sous la forme d'un vecteur (matières diffuses) ou d'une matrice (matières *glossy*) en chaque sommet. L'opérateur de transport représenté est celui de la source lumineuse vers la radiance incidente. Lors du rendu, la source lumineuse est projetée sur la base des harmoniques sphériques, puis elle est transformée pour chaque point de l'objet avec le vecteur (un produit scalaire) ou la matrice pour obtenir un éclairage de l'objet tenant compte de lui même. La radiance incidente est ensuite directement utilisable par la méthode de Kautz.

Il est possible d'utiliser un environnement dynamique pour déterminer la radiance incidente. Sloan réalise un rendu de l'environnement avec 6 images, puis le projette pour pouvoir l'utiliser. La projection n'est pas réalisable par la carte graphique, ce qui impose un traitement par le CPU et limite la taille des images. On notera que ce n'est pas un réel problème avec les environnements utilisés.

Cette méthode propose des effets très intéressants, mais son utilisation impose beaucoup de contraintes : le changement de repère des coefficients, la transformation pour obtenir la radiance incidente pour les matières *glossy* n'est pas réalisable sur le GPU et elle est très coûteuse ; l'objet ne peut pas être déformé, on ne peut lui appliquer que des transformations rigides ; la matière de l'objet ne peut pas être changée pendant l'exécution lorsque l'on utilise des réflections internes, car la matière influence les transferts de radiance pour cet effet ; et il faut aussi tenir compte des limitions liées à l'utilisation des harmoniques sphériques.

On remarquera que la transformation de la radiance incidente pour les matières diffuses peut maintenant être réalisée par fragment.

Analyse en composantes principales par morceaux et transfert de radiance précalculé

L'approche présentée par Sloan *et al.* dans [30] nécessite le stockage d'un grand nombre de vecteurs ou de matrices. Dans [29], Sloan *et al.* compressent les données utiles et améliore les performances de sa méthode.

Sloan utilise une analyse en composantes principales par morceaux [11] sur les vecteurs ou matrices de transport et effectue le rendu en utilisant l'approximation obtenue.

Contrairement à ce qu'il avait précédemment proposé, Sloan représente le transport de la source lumineuse vers la radiance émise. Ceci empêche la modification de la BRDF lors du rendu car elle est incluse dans la fonction, mais elle est nécessaire pour réaliser une bonne approximation.

Cette analyse regroupe les échantillons proches dans des blocs puis effectue une analyse en composantes principales. Elle permet une représentation efficace des vecteurs ou des matrices par quelques coordonnées dans la nouvelle base du bloc (un ensemble de vecteurs constants ou de matrices constantes dans notre cas). Ces coordonnées seront stockées en chaque sommet comme les vecteurs ou les matrices.

Le rendu se déroule comme ceci : dessiner le maillage dans le tampon de profondeur ; activer le *blending* en mode additif ; puis pour chaque bloc, calculer la multiplication des constantes du bloc avec les coefficients de la source de lumière sur CPU, les passer au GPU et dessiner les triangles du bloc, calculer la radiance émise en utilisant les coordonnées (une simple combinaison linéaire).

La précédente méthode permet un calcul de l'éclairage par fragment pour les surfaces diffuses. Celle qui suit le permet aussi pour les matières *glossy* car le GPU ne doit plus manipuler des matrices dans ce cas, mais des vecteurs.

Les triangles dont les sommets sont partagés entre plusieurs blocs sont dessinés plusieurs fois. Les blocs générés par l'analyse ne tiennent pas compte de la répartition des échantillons (les vecteurs ou les matrices) sur l'objet, ce qui augmente le nombre de fois que l'on dessine chaque triangle. Pour réduire cela, une partie des échantillons est changée de bloc et certains blocs sont réunies (optimisation entre erreur ajoutée et gain de cohérence).

Sloan *et al.* améliorent son approche de manière significative. Il propose une extension aux objets déformables [31] et aux objets rigides utilisant du *normal mapping* [28]. Kristensen *et al.* [16] étendent le travail de Sloan *et al.* aux sources de lumière proches. Pour réaliser cela, il introduit la notion de nuage de lumière (un ensemble dense de lumières ponctuelles réparties sur toute la scène) et adapte le travail de Sloan *et al.* à ce nouvel environnement lumineux (regroupement des points lumineux ayant un effet similaire sur la scène, découpage de la scène en zones pour améliorer le regroupement...). Cette méthode permet de placer des

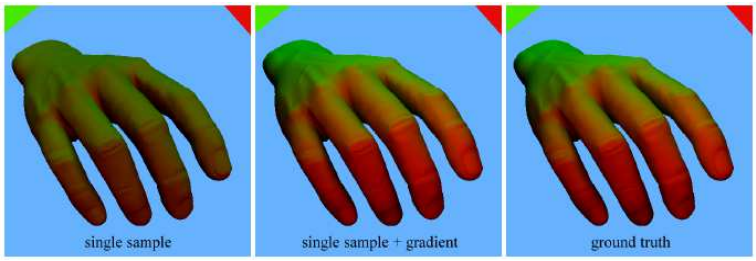

FIG. 1.7 – Cette figure montre un exemple de rendu avec la méthode de Annen *et al.*. L'image de gauche est obtenue avec la méthode de Sloan *et al.*. L'image de centre est obtenue avec la méthode de Annen *et al.*. L'image de gauche est l'image de référence. Ces images sont extraites de son article.

nouvelles lumières en modifiant la contribution des regroupements qui en sont proches, mais elle ne permet pas d'utiliser une scène dynamique. Annen *et al.* [1] proposent une solution pour utiliser un éclairage assez proche 1.7. L'éclairage est représenté par une série de coefficients en plusieurs points de l'objet. Puis lors du rendu, ces séries de coefficients sont combinées en tenant compte de leurs gradients (approximation au premier ordre) pour réaliser l'éclairage. La solution proposée par Annen est particulièrement intéressante mais reste limitée aux environnements lumineux composés de faibles fréquences. De plus, les sources très proches ne sont pas correctement supportées car l'estompage du à $\frac{1}{\|x-y\|^2}$ de la fonction G prend trop d'importance.

1.2.5 Précalculs et ondelettes

Les approches utilisant les harmoniques sphériques ne permettent pas de faire des rendus avec n'importe quel éclairage ou matière pour les objets. Elles ont tendance à lisser l'éclairage. Des méthodes à base d'ondelettes de Haar n'ayant pas cette limitation ont été proposées.

Ng *et al.* [21] proposent un algorithme utilisant les ondelettes de Haar. Comme les harmoniques sphériques, l'utilisation des ondelettes de Haar permet de réduire l'équation du rendu à un produit scalaire [21][22] entre les coefficients représentant l'opérateur de transport (incluant la BRDF) et ceux représentant la radiance incidente. Comme Sloan, Ng se limite à l'utilisation d'une partie des coefficients obtenus après projection et il réalise aussi un précalcul de l'opérateur de transport basé sur une simulation d'éclairage. L'intérêt des ondelettes de Haar est que l'élimination des coefficients faibles lors la reconstruction (l'intégration de l'équation du rendu) implique une erreur faible. Ceci permet une sélection intelligente des coefficients. Il devient possible de conserver les hautes fréquences qui contribuent à l'éclairage (voir figure 1.8). Ceci n'est pas réalisable avec une

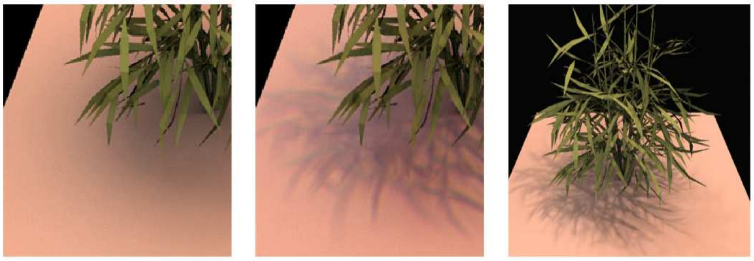

FIG. 1.8 – Cette figure illustre l'intérêt des ondelettes de Haar par rapport aux harmoniques sphériques pour un phénomène de hautes fréquences (ombres marquées par exemple). À gauche, l'image obtenue par la méthode de Sloan avec 25 coefficients. Au centre, l'image obtenue par la méthode de Ng avec 25 coefficients. À droite, l'image de référence. Ces images sont extraites d'un article de Ng.

base d'harmoniques sphériques, qui impose l'utilisation des premiers coefficients. Ng *et al.* proposent plusieurs politiques de sélection. La première consiste à conserver les coefficients élevés. Cette approche réduit l'erreur "globale" mais pas nécessairement l'erreur "visuelle". Ils proposent donc de tenir compte de l'opérateur de transport pour sélectionner les coefficients intéressants.

La solution de Ng *et al.* est interactive mais pose néanmoins un problème, l'utilisation d'une BRDF quelconque impose de fixer le point de vue contrairement à la solution de Sloan *et al.*. Lorsque la BRDF est diffuse le point de vue est modifiable. Ng *et al.* améliore son approche dans [22] au prix d'une grosse baisse des performances (plusieurs secondes par image à la place d'un rendu temps réel). Contrairement aux méthodes de Sloan *et al.* ou dérivées, les transformations appliquables aux objets sont restreintes aux translations et aux homothéties. On notera que ces approches utilisent massivement le CPU pour réaliser leurs calculs.

1.3 Bilan

Les différentes méthodes abordées utilisent soit des lumières ponctuelles soit un éclairage distant. De plus le dynamisme de la scène est souvent très limité. En effet, l'éclairage est souvent fixé, les objets non déformables et les matières limitées à un modèle spécifique.

Les méthodes préfiltrant une *environment map* souffrent des faibles capacités des GPU pour faire des convolutions et limitent leur utilisation aux environnements lumineux statiques pour les matières peu brillantes. Pour des matières très brillantes où une scène dynamique est envisagable, une méthode similaire à celle de Szirmay-Kalos *et al.* [32] utilisant une *environment map* de profondeur pour

limiter l'erreur due à un environnement proche est envisageable. Mais le procédé reste limité à des scènes restreintes et n'est utilisable que quelques fois par fragment.

Les méthodes utilisant une base d'harmoniques sphériques ou d'ondelettes de Haar permettent de capturer les différents effets d'illumination globale, mais ils dependent d'un long précalcul et ne sont pas utilisables avec des scènes dynamiques. Annen propose la méthode se rapprochant le plus de la thématique de ce stage (l'éclairage avec sources surfaciques proches). Il propose d'utiliser le gradient associé aux coefficients pour faire évoluer l'éclairage sur la surface mais son approche reste limitée environnements lumineux assez proches offrant peu de variations.

Ces différentes approches souffrent de plusieurs limitions assez rageantes. Elles permettent une visualisation satisfaisante des scènes mais ne permettent pas d'agir réellement sur la scène et elles demandent généralement de longs précalculs et le stockage de beaucoup d'informations. De plus, elles simplifient le problème en ne considérant qu'un éclairage distant ou des lumières ponctuelles. Lors de ce stage, nous avons tenu à traiter le cas des lumières surfaciques proches en gardant un maximum de dynamisme et en limitant les précalculs au minimum.

Le prochain chapitre sera consacré à la description du travail réalisé lors de ce stage.

Chapitre 2

Réalisation

Ce chapitre est consacré à la description du travail effectué lors de ce stage. Dans un premier temps, nous en délimiterons le cadre. Puis nous expliquerons pourquoi certaines approches utilisées précédemment ne sont pas valables. Enfin nous détaillerons les différentes approches testées pendant ce stage.

2.1 Cadre du stage

Nous avons choisi de travailler avec des sources de lumière placées près des objets et de ne prendre en compte qu'un éclairage direct. Les luminaires sont donc les seules surfaces à émettre de la radiance. Les effets comme les réflections internes, les caustiques ou les ombres n'entrent donc pas dans le cadre de ce stage.

Dans ces conditions, nous pouvons formuler l'équation du rendu 1.9 comme ceci :

$$L_o(x,v) = \sum_i \left(\int_{l_i} f_r(l^l, v^l) G(x,y) V_i(x,y) L_{l_i}(y,-l) dA_y \right) \tag{2.1}$$

l_i représentant la surface du $i$ème luminaire, V_i la visibilité entre un point x de la surface éclairée et un point y de l_i. Dans notre cas, la visibilité ne prend pas en compte les différents objets de la scène, mais juste le placement du luminaire.

Lors de l'état de l'art, nous avons décrit des méthodes dont certaines idées peuvent paraître applicables à notre problème. Nous allons expliquer pourquoi ce n'est pas le cas.

Toutes les mesures de performances sont réalisées une scène composée d'une source et d'un plan avec une résolution de 800 par 600 sur ordinateur équipé d'une GeForce 7800GT.

2.2 Approches caduques

Dans cette partie nous décrirons rapidement quelques approches envisagées et nous expliquerons pourquoi elles ne sont pas efficaces. La première approche part sur l'idée d'un précalcul et d'une tabulation de l'équation 2.1, la deuxième sur l'utilisation des harmoniques sphériques pour faciliter l'intégration.

2.2.1 Précalculer l'équation du rendu

L'équation 2.1 dépend de x (2 paramètres), de v (2 paramètres), du placement de chaque source de lumière (une orientation et une une position relative à x, 5 paramètres) soit une fonction 9D dans le cas ou l'on fixe la BRDF (ou SBRDF) des objets. La principale difficulté provient de la proximité des sources de lumières. Cette proximité donne beaucoup d'importance à G. Il est néanmoins possible de réduire le nombre de dimensions de cette fonction.

Une solution est de fixer la position des luminaires par rapport à la scène et d'effectuer un précalcul de l'équation 2.1 pour chaque sommet de la scène sur l'ensemble des points de vue possibles. Les calculs seront alors stockables sous la forme d'une *environment map* par sommet. Quelles que soient la méthodes de compression pour réduire le stockage des précalculs, le manque d'intérêt de ce genre d'approches vient de l'aspect statique des scènes visualisées. En effet, la solution décrite permet juste un changement de point de vue.

2.2.2 Harmoniques sphériques

Nous avons pu voir que l'utilisation des harmoniques sphériques pour représenter la BRDF et l'éclairage réduit l'équation du rendu à un produit scalaire. De plus il est possible de déduire les coefficients d'une fonction aisément à partir des coefficients d'une autre si elles sont liés par une rotation ou une homothétie [34]. Dans le cadre du stage, l'éclairage de deux points différents de la surface nécessite deux ensembles de coefficients qui ne sont ni liés par une rotation ni une homothétie (voir figure 2.1) et nous voulons traiter des scènes dynamiques avec des sources proches. Il est donc nécessaire de projeter la luminance incidente pour chaque sommet ou une grande partie en utilisant l'approximation de Annen *et al.* [1]. Déterminer le coefficient associé à une harmonique revient à calculer une intégrale [25]. Pour les lumières sphériques, il existe une formule analytique assez concise [30], sinon on est obligé de réaliser une intégration numérique.

L'intégration numérique nécessaire pour déterminer un coefficient est aussi coûteuse que l'intégration de l'équation du rendu lorsque l'environnement lumineux nécessite une *environment map* de bonne résolution. De plus nous savons

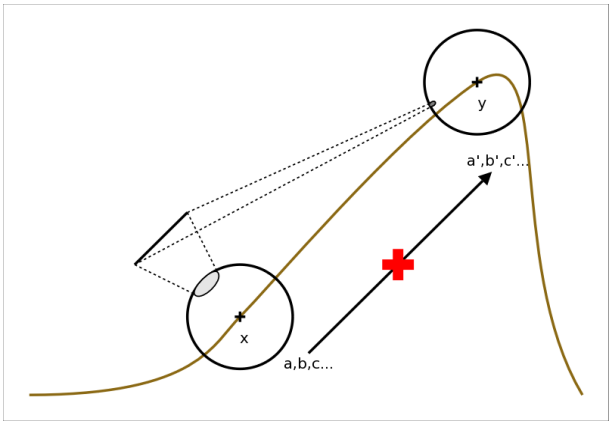

FIG. 2.1 – La courbe marron représente la surface. x et y sont deux points de cette surface. Les cercles centrés sur ces deux points représentent leurs éclairages. a,b,c et a',b',c' sont les deux séries de coefficients. Cette figure permet de voir que les fonctions décrivant l'éclairage des deux points ne sont pas liées par une rotation ou une homothétie et donc qu'il n'est pas possible d'obtenir a',b',c' à partir de a,b,c

que le nombre de coefficients nécessaire est assez important même dans le cas d'environnements lumineux composés de faibles fréquences.

Nous avons donc choisi de travailler sur l'intégration de l'équation du rendu directement car cette approche semble plus efficace dans pour des sources surfaciques proches et des scènes dynamiques. Dans les parties suivantes, nous détaillerons et discuterons les différentes approches testées.

2.3 Approche brutale

La première approche envisagée a été de résoudre directement l'équation du rendu sur GPU. Nous avons fait quelques hypothèses simplificatrices supplémentaires.

Ne considérer que des matières homogènes :

$$L_o(x,v) = \sum_i \left(\int_{l_i} f_r(l^l, v^l) G(x,y) V_i(x,y) L_{l_i}(y,-l) dA_y \right) \tag{2.2}$$

Des luminaires plans, ce qui permet d'écrire la fonction de visibilité V comme :

$$V_i(x,y) = V_{1_i}(x) V_{2_i}(x,y) \tag{2.3}$$

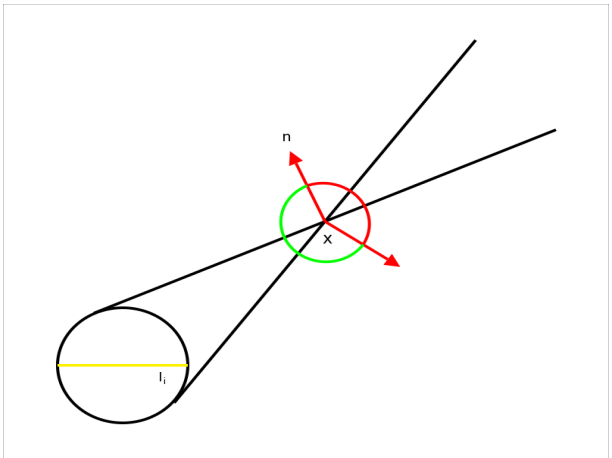

FIG. 2.2 – Lorsque la normale au point x se trouve dans la zone rouge, le luminaire ne contribue pas à l'éclairage.

avec

$$V_{1_i}(x) = \begin{cases} 1 & \text{si } x \text{ derrière le luminaire} \\ 0 & \text{sinon} \end{cases} \tag{2.4}$$

et

$$V_{2_i}(x,y) = \begin{cases} 1 & \text{si } l \cdot n > 0 \\ 0 & \text{sinon} \end{cases} \tag{2.5}$$

V_{1_i} peut être améliorée en considérant la sphère englobante du luminaire i (voir figure 2.2).

On obtient :

$$L_o(x,v) = \sum_i \left(V_{1_i}(x) \int_{l_i} f_r(l^l, v^l) G(x,y) V_{2_i}(x,y) L_{l_i}(y,-l) dA_y \right) \tag{2.6}$$

Des luminaires parfaits (radiance émise constante), ce qui donne :

$$L_o(x,v) = \sum_i \left(V_{1_i}(x) L_{l_i} \int_{l_i} f_r(l^l, v^l) G(x,y) V_{2_i}(x,y) dA_y \right) \tag{2.7}$$

Nous avons réalisé une intégration numérique par rectangle de l'équation 2.7 dans un *fragment shader*. On se contente de placer régulièrement un nombre constant d'échantillons sur chaque luminaire quel que soit le point observé de la surface. Cette intégration permet d'écrire l'équation 2.7 comme ceci :

$$L_o(x,v) \approx \sum_i \left(\frac{V_{1_i}(x) L_{l_i} A_{l_i}}{N} \sum_j^N f_r(l^l, v^l) G(x, S_i(j)) V_{2_i}(x, S_i(j)) \right) \tag{2.8}$$

La fonction $S_i(j)$ permet de placer régulièrement le $j^{\text{éme}}$ échantillon sur l_i. l est la direction de x à $S_i(j)$. A_{l_i} est l'aire du luminaire l_i.

Nous avons choisi de représenter un luminaire par un carré positionné et orienté arbitrairement dans l'espace. Ce choix permet une implémentation simple et peu coûteuse du placement régulier des échantillons sur le luminaire. En effet pour placer $n^2 = N$ échantillons, connaissant les positions de 3 coins de la lumière, il suffit de faire :

```
/*
    avec c1 : coin bas gauche
         c2 : coin bas droit
         c4 : coin haut gauche
*/
...
for(float u = 0.5 / n; u < 1.0; u += 1.0 / n)
    for(float v = 0.5 / n; v < 1.0; v += 1.0 / n)
{
    vec3 p = c1 + u * (c2 - c1) + v * (c4 - c1);
    ...
}
...
```

De plus cette forme particulière s'observe bien dans les reflets. En effet, la forme des reflets causée par ces luminaires est nettement différente de celle causée par un point de lumière, contrairement à celle provoquée par un luminaire sphérique. De plus, il est possible de représenter n'importe quelle forme plane en plaquant une texture (un texel blanc pour le luminaire, un texel noir sinon) :

```
...
/* On ne connaît pas le nombre d'échantillons avant de faire le calcul. */
int N = 0;

for(float u = 0.5 / n; u < 1.0; u += 1.0 / n)
    for(float v = 0.5 / n; v < 1.0; v += 1.0 / n)
{
    /* Est ce que l'échantillon tombe sur le luminaire ? */
    if(texture2D(textureForme,vec2(u,v)).r < 0.5)
```

```
    {
        /* Non */
        continue;
    }

    N++;

    /* Oui */
    vec3 p = c1 + u * (c2 - c1) + v * (c4 - c1);
    ...
}
...
```

L'échantillonnage d'un luminaire texturé dépend fortement du motif contenu dans la texture. Cette approche ne prend pas en compte la forme réelle du luminaire. Dans la suite de ce rapport nous ne traiterons que de luminaires carrés d'une largeur donnée.

Le problème principal de cette approche est de fixer un nombre d'échantillons suffisamment élevé pour que le calcul paraisse correct tout en conservant de l'interactivité. Notre intégration revient à considérer une source surfacique comme un ensemble de lumières ponctuelles régulièrement espacées. Un manque d'échantillons fais apparaître ces lumières ponctuelles imaginaires sous la forme de halos (voir figure 2.3). Nous proposons donc une première solution pour adapter l'échantillonnage.

2.4 Amélioration de la première approche

Partant de l'observation faite précédemment, nous avons entrepris d'étudier le nombre d'échantillons nécessaires pour réaliser une bonne approximation avec l'approche brutale.

2.4.1 Visualiser le problème

La figure 2.4 permet de visualiser ce problème dans un cas simple. On remarque que deux halos vont se former autour des points A et C alors que le point B ne sera presque pas éclairé. Un échantillonnage plus important réduit ce problème comme le souligne la figure 2.4. L'approximation est satisfaisante quand on ne

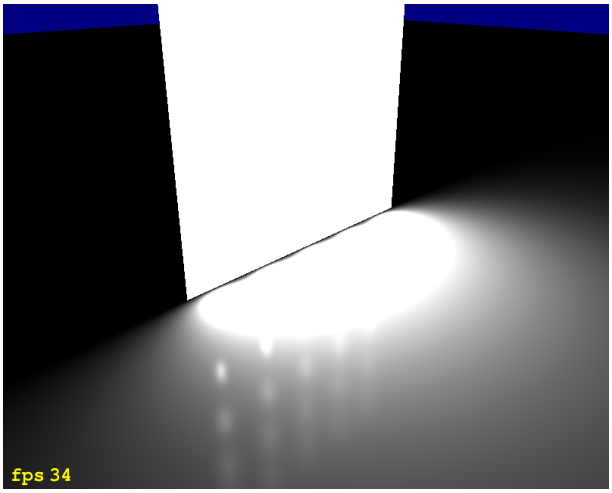

FIG. 2.3 – Effets du manque d'échantillons.

perçoit plus ces oscillations dans les reflets. On remarque aussi que les oscillations sont plus visibles sur les bords des reflets. En effet une lumière puissante aura tendance à saturer le centre de son reflet.

2.4.2 Déterminer l'échantillonnage nécessaire

Afin de déterminer l'échantillonnage nécessaire, nous avons recherché des fonctions de prédiction à partir de mesures empiriques. Pour réaliser ces mesures, nous avons tenu compte de plusieurs observations.

La figure 2.5 met en évidence l'effet de l'orientation du luminaire par rapport au sol. De plus il existe un lien entre la position et la taille de la source comme le montre la figure 2.6. L'échantillonnage nécessaire dépend de l'angle solide de la source. Ces observations permettent de limiter les mesures à une seule lumière (carré de 1 unité de côté) en faisant varier la distance au point observé pour faire varier l'angle solide. Ce choix permet de faire les mesures plus facilement et ainsi d'obtenir de meilleurs résultats. Le phénomène de saturation doit aussi être pris en compte. La figure 2.4 met en avant le rôle que tient la BRDF dans le choix de l'échantillonnage. Il aurait été meilleur pour un lobe « moins fin ». Nous avons donc étudié plusieurs types de BRDF, le modèle de Lambert, le modèle de Phong et celui de Blinn-Phong.

Pour réduire le nombre de mesures à réaliser ainsi que la complexité de la fonction de prédiction, nous n'avons pas tenu compte de certains paramètres en choisissant leurs pires configurations lors des mesures. Cette démarche vise à maximiser l'erreur visuelle et ainsi à obtenir une méthode conservatrice. Nous allons décrire la prise des mesures et discuter des résultats obtenus pour le modèle

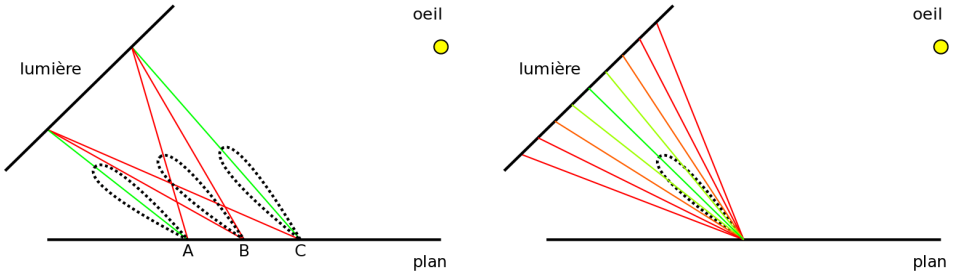

FIG. 2.4 – En pointillés, une BRDF possédant un seul lobe. Sur le sol, trois points A, B et C ou nous évaluons notre équation. Les traits colorés reliant la lumière au sol matérialisent l'échantillonnage. Un trait vert signifie que la valeur de la BRDF pour l'échantillon est élevée, un trait rouge qu'elle est faible. La variation de la valeur est représentée par un dégradé du vert au rouge.

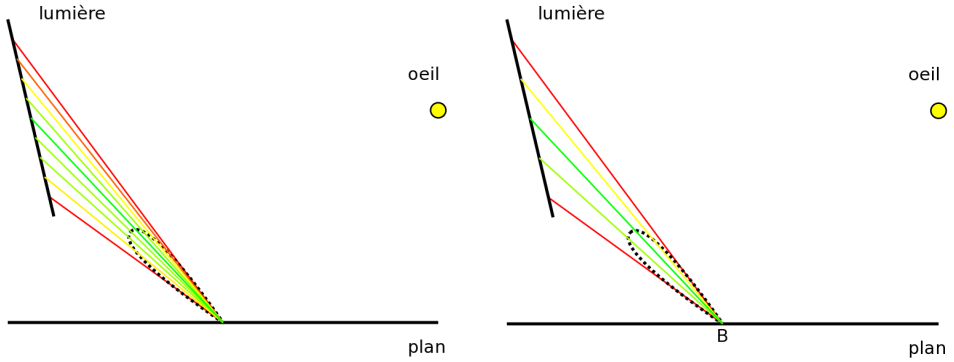

FIG. 2.5 – On considére que l'échantillonage représenté sur la deuxième image de la figure 2.4 est suffisant. Sur l'image de gauche, on remarque que le nombre d'échantillons est trop élevé (angle entre les échantillons trop faible). L'image de droite montre un échantillonage suffisant pour cette scène.

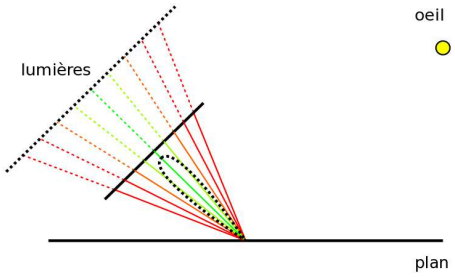

FIG. 2.6 – Deux luminaires de taille différentes nécessitant le même échantillonage.

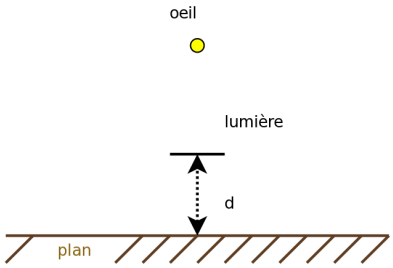

FIG. 2.7 – Un point jaune représente l'oeil de l'utilisateur, une ligne hachurée marron le plan illuminé, une ligne noire le luminaire et la flèche en pointillés la distance que l'on fait varier.

de Lambert, le modèle de Phong et celui de Blinn-Phong.

Modèle de Lambert

Nous avons ajusté visuellement pour différent échantillonnage (N^2 échantillons pour $N \in [1,50]$) la distance limite à laquelle les résultats obtenus sont visuellement convaincants. Les mesures pour ce modèle ont été réalisée sur la scène de la figure 2.7.

Elle est composée d'un plan et d'un luminaire « unité »qui évolue verticalement face au plan. La matière du plan est diffuse. Le luminaire a été placé face au plan pour pouvoir mieux apprécier l'erreur. En effet, cette configuration réduit l'effet d'estompage du à la fonction G. La position de la caméra n'est pas importante, mais la placer au dessus permet une meilleur visualisation. La puissance du luminaire est réglée pour que le centre du reflet soit juste saturé.

La courbe obtenue fournie de bonnes informations. Un seul échantillon suffit dès que la distance à la source lumineuse est approximativement égale à sa largeur. De plus elle est quasiment linéaire et elle est approchable efficacement (voir figure 2.8) par la fonction :

$$\Delta_{lambert}(d) = \begin{cases} 0.727d + 0.055 & \text{si } d < 1.3 \\ 1 & \text{sinon} \end{cases} \tag{2.9}$$

$\Delta_{lambert}$ donne la distance nécessaire entre deux échantillons successifs en fonction de la distance du luminaire par rapport au point éclairé. Le pas d'échantillonnage calculé par $\Delta_{lambert}$ ne donne pas nécessairement un échantillonnage régulier. La valeur calculée est donc ramenée au plus proche pas inférieur, ce qui peut être fait avec une texture 1D (astuce notamment utilisée pour faire du rendu *cartoon*).

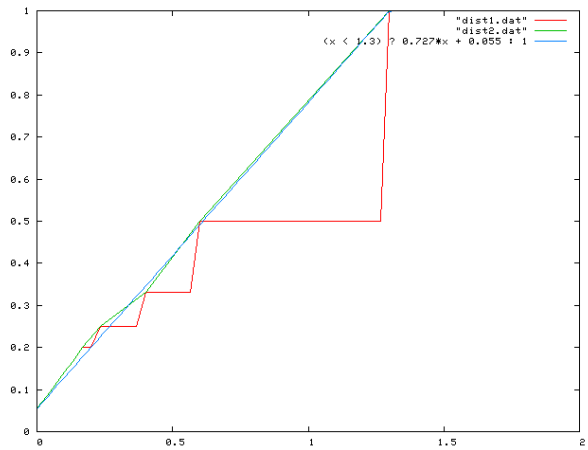

FIG. 2.8 – La courbe rouge représente toutes les mesures, son aspect crénelé provient de la régularité de l'échantillonnage. La courbe verte représente la courbe approchant les mesures (fonction $\Delta_{lambert}$).

On notera que la contribution d'une partie diffuse est calculable exactement pour des sources de lumière polygonales [2], mais que ce calcul est relativement coûteux et rapidement inutile.

Modèle de Phong

Le modèle de Phong ajoute une partie spéculaire au modèle de Lambert. Quel que soit le point de vue, le lobe conserve la même forme et il est symétrique par rapport à l'axe de réflection. Ce comportement particulier nous permet de fixer l'orientation du luminaire perpendiculairement à cet axe. Une solution simple consiste à reprendre la scène utilisée pour le modèle de Lambert (bon pour G) et à remplacer le modèle de Lambert par un modèle de Phong sans partie diffuse ($\rho_d = 0$). L'exposant influe sur la forme du lobe. Nous tenons donc compte de ce paramètre dans nos mesures.

Les mesures ont été faites différemment que pour celles du modèle de Lambert pour lesquelles l'aspect crénelé de f ne pause pas de problème, en effet on peut trouver une fonction l'approchant « à la main ». La surface obtenue pour le modèle de Phong nécessite plus de précaution. Une optimisation manuelle n'est pas réalisable. Il faut donc éviter le crénelage pour faire une optimisation efficace. La solution adoptée est de ne plus se limiter aux pas donnant des échantillonnages *corrects* ($1, \frac{1}{2}, \frac{1}{3} \ldots$), mais de régler le pas d'échantillonnage librement. Cette solution ne permet pas une comparaison avec un éclairage de référence, mais donne plus de liberté dans le choix de l'échantillonnage. Le manque d'échantillons provoque une oscillation du reflet avant de laisser apparaître des halos lumineux. Une

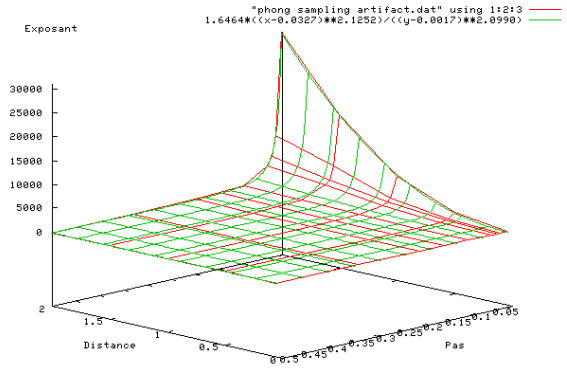

FIG. 2.9 – La surface rouge est mesurée, la verte est son approximation.

erreur trop importante a donc été associée avec l'apparition de cette oscillation. Sachant que l'échantillonnage nécessaire décroît très vite, nous avons limité nos mesures à plusieurs courtes distances (0.05, 0.5, 1, 1.5 et 2 fois la largeur du luminaire) puis pour chaque échantillonnage nous avons recherché l'exposant limite jusqu'auquel il convient.

La surface obtenue est réglée (voir figure 2.9) ce qui a permit de trouver la forme de Δ_{phong} à optimiser et la fonction de coût associée très rapidement.

$$\Delta_{phong}(d,s) = \frac{a_1(d+a_2)^{a_3}}{(s+a4)^{a_5}} + a_6 \tag{2.10}$$

Nous avons utilisé la méthode des moindres carrés non linéaire fournie par Matlab pour minimiser notre fonction de coût. Voici la surface obtenue (voir figure 2.9) :

$$\Delta_{phong} f(d,s) = \frac{1.6464(d-0.0327)^{2.1252}}{(s-0.0017)^{2.0990}} \tag{2.11}$$

d est la distance et s est le pas d'échantillonnage.

La fonction f donne l'exposant maximal utilisable en fonction de la distance et de l'échantillonnage. Elle est bijective et peut être inversée sans problème pour donner le pas d'échantillonnage en fonction de l'exposant et de la distance :

$$\Delta_{phong}(d,e) = \left(\frac{1.6464(d-0.0327)^{2.1252}}{e}\right)^{0.4761} - 0.0017 \tag{2.12}$$

d est la distance et e est l'exposant. Une fois le pas calculé, il est corrigé avec la même méthode que la partie diffuse.

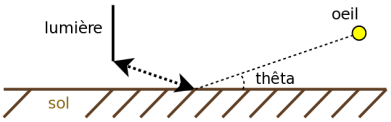

FIG. 2.10 – Un point jaune représente l'oeil de l'utilisateur, une ligne hachurée marron le plan illuminé, une ligne noire le luminaire et la flèche en pointillés la distance que l'on fait varier. Cette scène est utilisée pour faire les mesures pour le modèle de Blinn-Phong.

Tout comme le modèle de Lambert, on remarque que l'exposant maximal utilisable pour chaque échantillonnage augmente très vite avec la distance.

Modèle de Blinn-Phong

Tout comme le modèle de Phong, le maximum d'une BRDF obtenue à partir du modèle de Blinn-Phong se trouve sur l'axe de réflection. Par contre, le lobe s'amincit pour des angles rasants. Ce comportement rend les mesures plus délicates à effectuer. Pour obtenir une fonction de prédiction ne dépendant pas de θ_o ($\rho_o = (\theta_o, \phi_o)$, la pire configuration est un angle ($\theta_o \approx 0°$). La scène utilisée en tient compte (voir figure 2.10).

Nous avons choisi un angle $\theta_o = 5°$. Le luminaire évolue dans la direction de l'axe de réflection pour conserver le reflet au même endroit. La distance d'une mesure est déterminée comme étant la distance la plus courte entre le centre du reflet et la lumière. Ce calcul nous force à ne tenir compte du centre du reflet pour fausser les mesures le moins possible. Le reste de la procédure est similaire à celle du modèle de Phong.

On obtient cette fonction :

$$\Delta_{blinn}(d,e) = \left(\frac{2.6419(d+0.1364)^{2.0780}}{e}\right)^{2.4589} + 0.0082 \tag{2.13}$$

La surface obtenue pour le modèle de Blinn-Phong est similaire à celle du modèle de Phong. La sous partie suivante sera consacrée à la description de l'utilisation des trois fonctions de prédiction obtenues.

Mise en oeuvre

Afin d'utiliser nos fonctions, nous devons déterminer la distance la plus courte du point observé x à la source de lumière. Dans un premier temps, on projette x sur le plan support de la lumière. On utilise les points c_1,c_2 et c_4 pour former la

base du plan (voir équation 2.16) pour projetter x. On sait que les coordonnées d'un point appartenant à la lumière $\in [0,1]^2$. Si le point projetté n'appartient pas à la source, il suffit de prendre le plus proche point dans $[0,1]^2$ (voir équation 2.19). Puis, il reste à replacer ce point dans le repère courant (voir équation 2.20) pour la suite des calculs.

$$x_p = (x_x p, x_y p) \tag{2.14}$$

$$x_{xp} = \frac{(x - c_1) \cdot \overrightarrow{c_1 c_2}}{\|c_2 - c_1\|} \tag{2.15}$$

$$x_{yp} = \frac{(x - c_1) \cdot \overrightarrow{c_1 c_4}}{\|c_4 - c_1\|} \tag{2.16}$$

$$x_s = (x_{xs}, x_{ys}) \tag{2.17}$$

$$x_{xs} = min(max(0.0, x_{xp}), 1.0) \tag{2.18}$$

$$x_{ys} = min(max(0.0, x_{yp}), 1.0) \tag{2.19}$$

$$x_{sw} = c_1 + x_{sx}\overrightarrow{c_1 c_2} + x_{sy}\overrightarrow{c_1 c_4} \tag{2.20}$$

x_p est le point x projeté sur le plan contenant la source. x_s est le point de la source le plus proche de x_p. x_{sw} est le point x_s exprimé dans le repère courant (repère local ou repère de l'oeil par exemple).

La deuxième considération à prendre en compte est la justesse de nos fonctions de prédiction. Elles sont issues de mesures empiriques et manquent de précision. Pour compenser ceci, nous utilisons 95% de la valeur prédite pour $\Delta_{lambert}$, 90% pour Δ_{phong} et 50% pour Δ_{blinn}. La très faible valeur utilisée pour Δ_{blinn} s'explique par le comportement aux angles rasants évoqué précédemment. Nous limitons le pas d'échantillonnage à 0.05 « unité de luminaire ».

Toutes les transitions d'un pas d'échantillonnage à un autre introduit des artefacts visuels qui se manifeste sous la forme de paliers sur les reflets (voir figure 2.11). Une idée naturelle est d'utiliser un filtrage linéaire pour passer progressivement d'un échantillonnage à l'autre. Cette solution est trop coûteuse car elle demande de calculer deux échantillonnages pour chaque fragment. Nous avons donc utilisé un *dithering* pour brouiller les paliers.

La fonction de prédiction utilisée pour le modèle de Phong devient :

$$\Delta_{phong2}(d,e) = max(0.05, \Delta_{phong}(d,e) - 0.1x_i) \tag{2.21}$$

x_i est une valeur aléatoire uniformément distribuée dans $[0,1]$. Le bruit ajouté brouille efficacement la transition entre deux échantillonnages (voir figure 2.12).

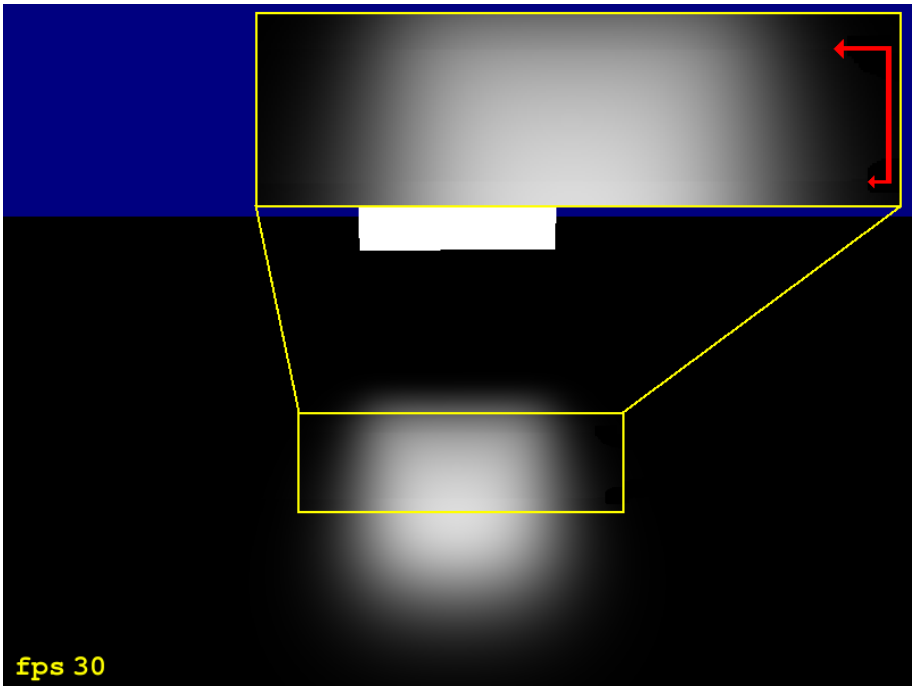

FIG. 2.11 – Résultat en utilisant la fonction Δ_{phong} qui ne supprime pas les artefacts visuels. Nous pouvons voir les paliers apparaître lors du changement d'échantillonnage (en face des flèches rouges).

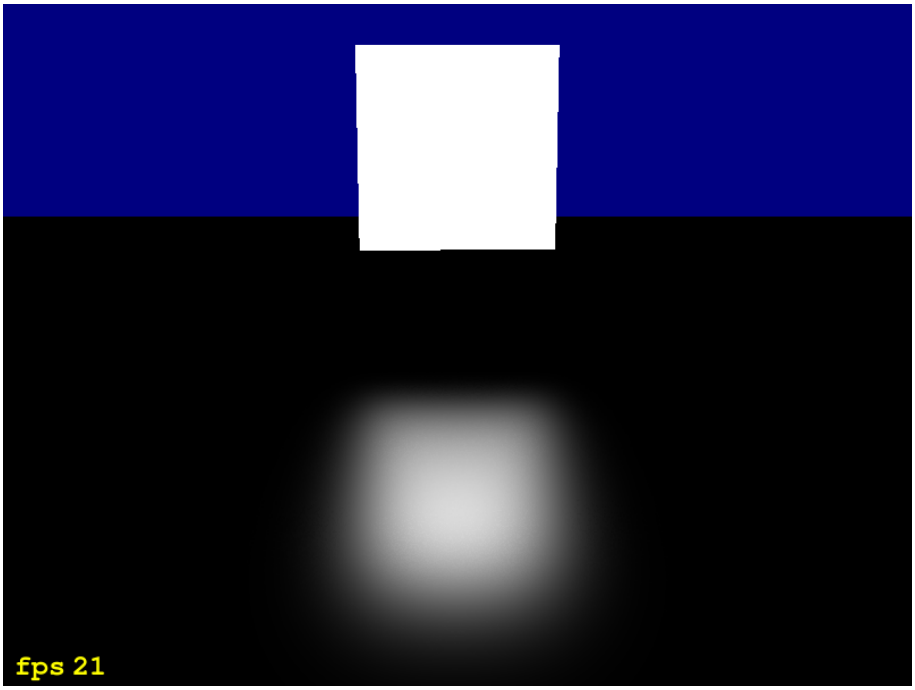

FIG. 2.12 – Résultat en utilisant la fonction Δ_{phong2} qui supprime les artefacts visuels.

Il permet aussi de tenir plus compte de la valeur prédite (100% dans le cas de Δ_{phong}), mais réduit considérablement les performances. Les fonctions $\Delta_{lambert}$ et Δ_{blinn} sont modifiables de manière similaire.

Le nombre d'échantillons nécessaires pour la partie diffuse des modèles de Phong et de Blinn-Phong est toujours inférieur à celui de leur partie spéculaire. Pour éviter les calculs inutiles, nous ne separons pas les deux parties lors du rendu et nous utilisons seulement Δ_{phong2} ou son homologue pour le modèle de Blinn-Phong.

L'étude effectuée sur ces trois modèles nous permet de réduire très rapidement le nombre d'échantillons. Dans la prochaine sous partie nous comparerons l'approche brutale et son amélioration.

2.4.3 Comparaison entre la méthode brutale et la méthode adaptative

Le calcul de ces deux méthodes est réalisé dans un *fragment shader*, donc au moins une fois par pixel. Le taux de rafraîchissement varie en fonction du

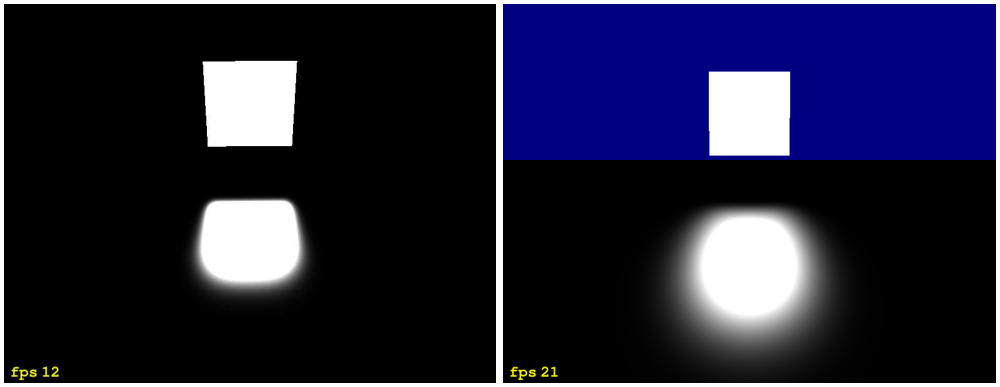

FIG. 2.13 – Avec la première méthode, faire varier le point de vue modifie du simple au double les performances (12 *fps* contre 21).

nombre de pixels traités à l'écran. Pour une même méthode avec la même scène, le nombre d'images par seconde (*fps*) peut varier énormément (voir figure 2.13). Ainsi lorsque l'on compare deux scènes différentes, on peut obtenir des résultats très différents pour une même méthode. Pour la méthode adaptative ce phénomène est très marqué car il dépends des lumières utilisées, de leur position, des matières, du point de vue, du nombre de fragments traités. . .Les deux méthodes proposent un rendu temps réel dans certains cas, mais peuvent aussi terrasser la machine dans d'autres cas. On peut néanmoins faire quelques comparaisons pour une scène identique entre les deux méthodes.

La méthode adaptative n'apporte pas nécessairement un gain de performance (voir figure2.15). Lorsque l'échantillonnage nécessaire est faible (nombre d'échantillons ≤ 25), l'évaluation des fonctions de prédiction devient plus coûteuse que l'intégration elle même. Par contre, elle est plus efficace avec des matières très brillantes (voir figure 2.14) demandant un fort échantillonnage près de la source.

La méthode adaptative apporte une première amélioration, elle permet d'utiliser des matières plus brillantes. Nous avons remarqué que l'échantillonnage nécessaire est plus faible lorsque la source lumineuse est de coté. Ceci est directement lié à la diminution de l'angle solide. Partant des fonctions Δ_*, il est possible d'utiliser l'angle solide et l'exposant à la place de la distance et de l'exposant. En effet, les mesures ont été effectuées pour une distance et une orientation à partir duquel nous pouvons déterminer un angle solide. Cette solution permet de réduire l'échantillonnage pour certain point de vue, mais ne règle pas le principal problème de cette méthode. Dans une scène ou la lumière est proche, beaucoup d'échantillons restent inutiles avec la méthode brutale et la méthode adaptative. La prochaine approche propose une solution pour éliminer ces échantillons.

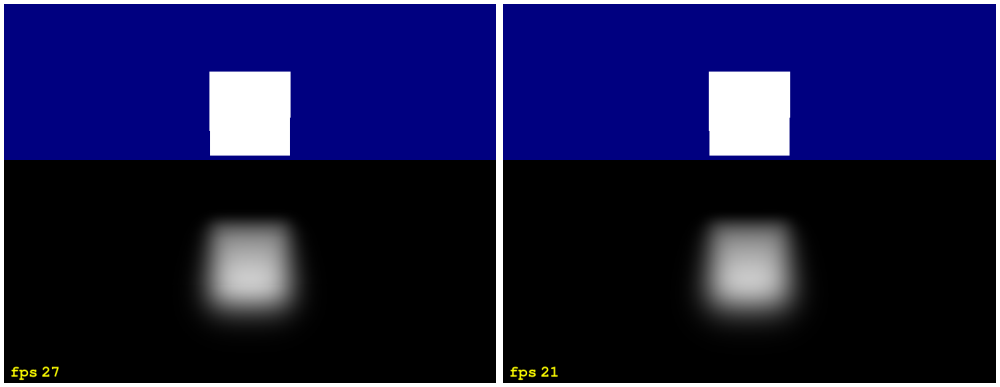

FIG. 2.14 – L'image de gauche correspond à la méthode adaptative (27 *fps*), la gauche à la méthode brutale (21 *fps*). Modèle de Phong avec $\rho_d = 0$, $\rho_s = 1$ et $e = 500$.

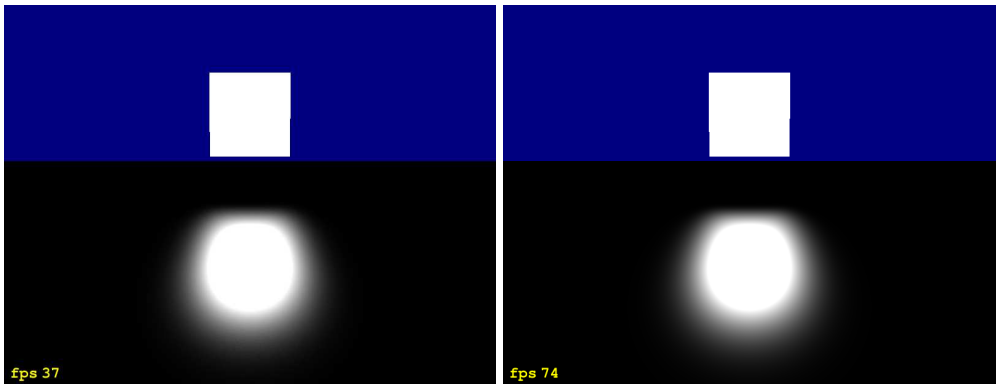

FIG. 2.15 – L'image de gauche correspond à la méthode adaptative (37 *fps*), la gauche à la méthode brutale (74 *fps*). Modèle de Phong avec $\rho_d = 0$, $\rho_s = 1$ et $e = 150$.

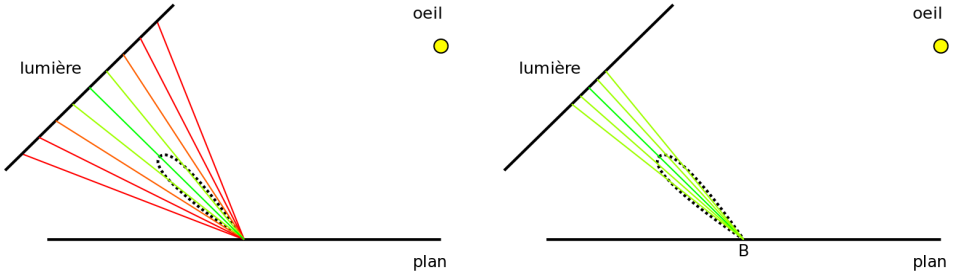

FIG. 2.16 – A gauche l'échantillonnage régulier des deux première méthodes. A droite celui que l'on souhaite avoir.

2.5 Approche biaisée

Cette seconde approche se concentre sur la BRDF pour résoudre l'équation 2.7, elle vise à ne pas échantillonner lorsque la valeur de la BRDF est faible (voir figure 2.16). On se limite à la partie spéculaire du modèle de Phong. Nous utilisons les mêmes simplifications (luminaires plans...) que pour la dernière approche.

Si on observe la partie spéculaire du modèle de Phong, on remarque que la fonction $cos^e(x)$ est presque nulle sur une grande partie de l'intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$ (voir figure 2.17). Partant de cette observation, nous avons cherché à réduire et à positionner l'intervalle d'intégration de chaque luminaire.

L'intervalle utile $[-a, a] \subseteq [-\frac{\pi}{2}, \frac{\pi}{2}]$ de la fonction $cos^e(x)$ est déterminé en résolvant :

$$t \cdot \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} cos^e(x)dx = \int_{-a}^{a} cos^e(x)dx \tag{2.22}$$

t est une tolérance arbitraire $\in [0, 1]$. Le choix de t doit être très précis pour réduire l'intervalle tout en garantissant que le biais commis n'est pas visuellement gênant. Nous avons choisi $t = 0.99$ après expérimentation (voir figure 2.18).

L'équation 2.22 pour $t = 0.99$ est précalculable pour $e \in [1, \approx 300]$ et approchable par (voir figure 2.19) :

$$a(e) = \frac{1.8026}{e^{0.2868}} - 0.2162 \tag{2.23}$$

L'intervalle $[-a, a]$ définit un cône plein orienté dont l'origine est le point observé x, l'axe est le vecteur réfléchi et l'angle ouverture est a. Il représente l'ensemble des directions intéressantes et peut être échantillonné régulièrement. Cette solution demande de tester chaque échantillons pour savoir s'il touche ou non une source de lumière, ce qui est inefficace. Une solution plus performante est de

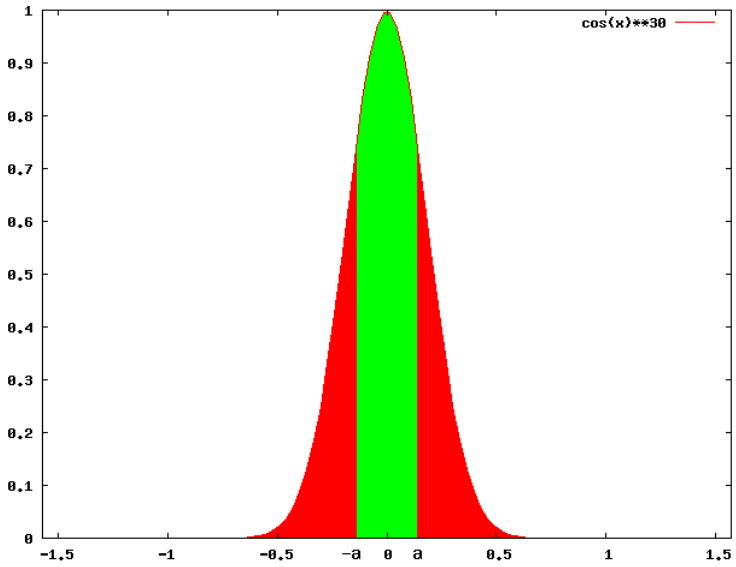

FIG. 2.17 – La fonction $cos^{30}(x)$ pour $x \in [-\frac{\pi}{2}, -\frac{\pi}{2}]$. a représente l'angle limite après lequel on ne veut plus échantillonner la BRDF. On remarque que les valeurs sont proches de zéro quand $|x| > a = \frac{\pi}{3}$ pour $e = 30$.

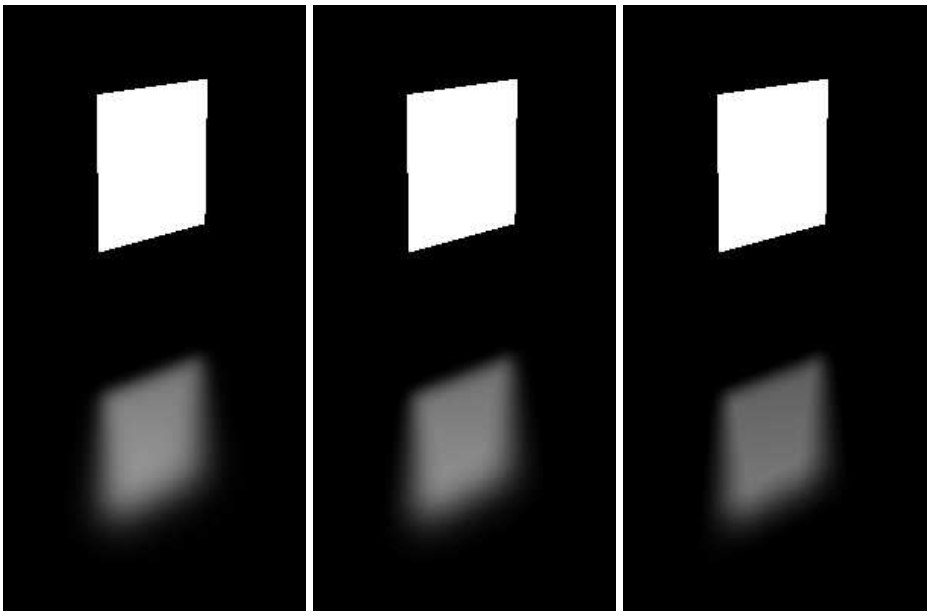

FIG. 2.18 – Rendu du modèle de Phong avec un exposant $e = 500$. À gauche, l'image de référence obtenue par la première méthode (3 *fps*). Au centre, l'image obtenue avec une tolérance $t = 0.99$ (250 *fps*). À droite, l'image obtenue avec une tolérance $t = 0.90$ (250 *fps*). La dernière image laisse apparaître le biais.

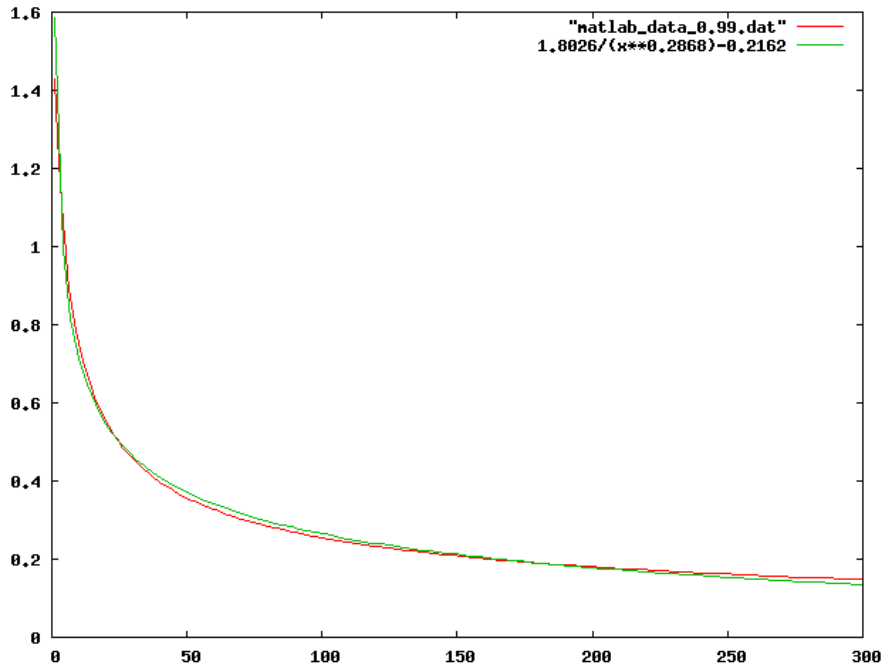

FIG. 2.19 – En rouge la fonction calculée. En vert la fonction approximée.

déterminer une approximation rectangulaire de l'intersection entre le cône et le plan du luminaire puis d'échantillonner cette zone régulièrement si elle existe.

Lorsqu'elle existe, l'intersection d'un cône orienté et d'un plan est une conique. Six cas peuvent se présenter, elle est soit une ellipse, soit une hyperbole, soit une parabole, soit un point, soit une ou deux droites. Les trois derniers cas apparaissent quand la base du cône se trouve sur le plan du luminaire. Dans cette position, peu importe le type d'intersection, l'éclairage du au luminaire sera nul. L'apparition des cas non dégénérés est visible sur la figure 2.20. Lors de l'étude de notre première méthode pour le modèle de Phong, on a remarqué qu'un luminaire échantillonné de coté nécessite moins d'échantillons. Ce cas intervient lorsque que l'intersection est une hyperbole ou une parabole. Partant de ces observations, nous proposons une approximation peu fidèle mais peu coûteuse qui permet une bonne intégration.

Lorsque nous sommes dans le cas d'une ellipse, nous cherchons une boîte englobante pour l'utiliser comme intervalle d'intégration (voir figure 2.21). Cette boîte englobante est centrée sur l'intersection du plan et du rayon représentant l'axe du cône. La demi largeur de la boîte L est calculée par :

$$L = \frac{d sin(\alpha)}{cos(\beta)} \tag{2.24}$$

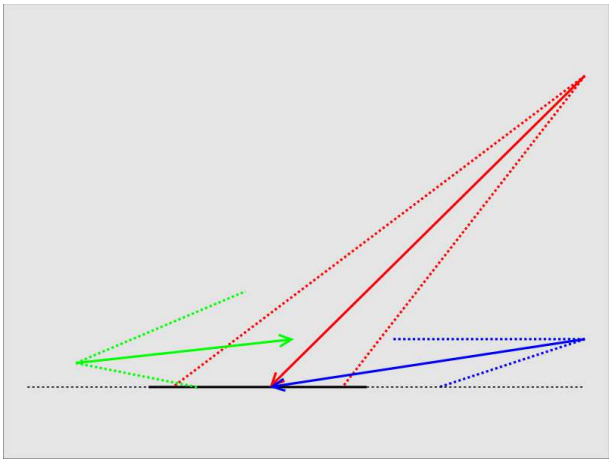

FIG. 2.20 – Le trait noir représente le luminaire. En pointillés, le plan du luminaire. En vert, un cône causant une hyperbole. En rouge, le cas de l'ellipse. En bleu, celui de la parabole. On remarque que les deux derniers cas apparaissent lorsque l'utilisateur se place sur le coté du luminaire.

Lorsque l'on passe du cas de l'ellipse au cas de la parabole puis de l'hyperbole, l'approximation de l'intersection sera quasiment toujours (sauf lumières très grandes collées à un sol spéculaire) la source en entier. Dans ce cas, nous choisissons directement la source complète comme intervalle d'intégration.

Dans le cas d'une hyperbole ou d'une parabole, la boîte englobante recouvre entièrement le luminaire.

Cette méthode permet un rendu fidèle de matières peu ou très brillantes (voir image centrale de la figure 2.18) en conservant un échantillonnage faible et constant, contrairement à la méthode adaptative dont l'échantillonnage varie en

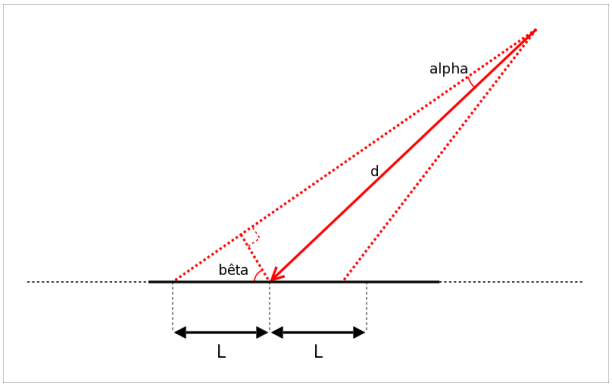

FIG. 2.21 – Cette figure illustre le calcul effectué pour obtenir la boîte englobante.

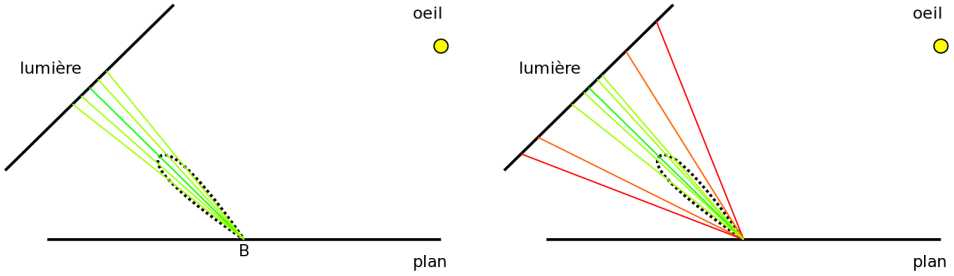

FIG. 2.22 – À gauche, l'échantillonage régulier de la méthode biaisée. À droite, celui de la méthode probabiliste.

fonction de la distance et de l'exposant. L'échantillonnage constant permet d'obtenir un taux de rafraîchissement plus stable qui ne dépend que du nombre de sources et du nombre pixels illuminés. Elle est donc plus efficace que la méthode adaptative.

Parallèlement à cette approche, nous avons travaillé sur une approche probabiliste que nous allons vous présenter dans la partie suivante.

2.6 Approche probabiliste

Comme l'approche précédente, l'approche probabiliste se concentre sur la BRDF pour résoudre l'équation 2.7. Elle vise à réduire le nombre échantillon contribuant peu au résultat et à augmenter le nombre de ceux qui y contribuent (voir figure 2.22). La différence entre les deux approches est que l'intégration probabiliste n'est pas biaisée. On se limite aussi à la partie spéculaire du modèle de Phong.

2.6.1 Intégration probabiliste

Les méthodes dites de Monte Carlo sont particulièrement utiles pour calculer des intégrales de dimension élevée. Nous allons en faire une description rapide.

Quelques rappels

Intéressons nous au cas simple d'une fonction $f : \mathbb{R} \to \mathbb{R}$:

$$I = \int_a^b f(x)dx \tag{2.25}$$

$$I \approx I_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \tag{2.26}$$

Avec

$$\int_{-\infty}^{\infty} p(x)dx = \int_a^b p(x)dx = 1 \tag{2.27}$$

Les échantillons x_i sont choisis selon la fonction de densité de probabilité p. On notera que l'estimateur I_N est non biaisé ($E[I_N] = I$), que sa convergence est en $N^{-\frac{1}{2}}$ et que sa variance est d'autant plus faible que p est proportionnelle à f [27].

Pour une fonction $f : \mathbb{R}^2 \to \mathbb{R}$:

$$I = \int_a^b \int_c^d f(x,y)dxdy \tag{2.28}$$

$$I \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i,y_i)}{p(x_i,y_i)} \tag{2.29}$$

Avec

$$\int_a^b \int_c^d p(x,y)dxdy = 1 \tag{2.30}$$

Générer des échantillons

Nous disposons généralement d'un tirage uniforme (x_i,$i = 1,2\ldots$) dans $[0,1]$. Lorsque nous voulons obtenir un tirage (x'_i,$i = 1,2\ldots$) selon une loi de distribution représentée par p, deux méthodes s'offrent à nous.

Inversion de la fonction de répartition Si on connaît la fonction de répartition P associée à p et qu'elle est inversible, il est possible de générer un tirage suivant p comme ceci :

$$x'_i = P^{-1}(x_i) \tag{2.31}$$

Lorsqu'il n'existe pas une solution analytique simple pour calculer et inverser P, une approximation numérique est utilisable. Les valeurs manquantes sont obtenues par interpolation linéaire.

Lorsque p et P ont deux paramètres (génération sur une surface par exemple), inverser P à partir d'un seul échantillon n'est plus possible. La solution décrite par Shirley [27] revient à décomposer p en deux fonctions q et q_x (fonction de

densité conditionnelle sachant x) étant elles mêmes des fonctions de densité de probabilité :

$$q(x) = \int_{-\infty}^{\infty} p(x,y)dy \tag{2.32}$$

$$q_x(y) = \frac{p(x,y)}{q(x)} \tag{2.33}$$

Une fois décomposée sous cette forme, il est possible d'utiliser la méthode précédente sur chaque partie pour utiliser une fonction de densité de probabilité ayant deux paramètres.

Méthode du rejet Une autre solution consiste à utiliser la méthode du rejet. Pour appliquer cette méthode, il faut pouvoir majorer p par une constante C et une nouvelle fonction de densité de probabilité q pratique (Q^{-1} facilement calculable...). Pour trouver une valeur aléatoire x_i valide, on utilise cet algorithme :

Algorithm 1 Méthode du rejet

repeat
  Générer u_1 uniformément dans $[0,1]$;
  $x_i = Q^{-1}(u_1)$;
  Générer u_2 uniformément dans $[0,1]$;
until $u_2 > p(x_i)/(Cq(x_i))$
Utiliser x_i ;

Prenons un exemple pour illustrer l'approche. On souhaite échantillonner uniformément un polygone quelconque. Utiliser cette méthode revient à déterminer un rectangle englobant, générer nos échantillons uniformément sur ce rectangle et ne garder que les échantillons tombant sur le polygone. Pour un échantillon généré sur le polygone, plusieurs échantillons générés sur le rectangle sont nécessaires. Cette méthode donne de mauvais résultats lorsque l'on fait une intégration [27], il faut donc l'utiliser seulement quand c'est nécessaire. La méthode du rejet est utilisable avec des fonctions de densité de probabilité à plusieurs dimensions.

2.6.2 Monte Carlo et équation du rendu

Lorsqu'on fait une intégration probabiliste de l'équation du rendu, deux types de stratégies d'échantillonnage s'offrent à nous. La première est d'utiliser une densité de probabilité proportionnelle à $f_r(x,v,l)$ ou à $f_r(x,v,l)\cos(\theta_x)$. La seconde est de considérer l'importance de la contribution des sources de lumières pour éviter d'échantillonner une petite, lointaine et faible source de lumière. Les stratégies

se basant sur la BRDF ont tendance à « rater »les sources de lumineuses puissantes mais de petite taille. Alors que celles se basant sur les sources de lumières ont tendance à « rater »les parties importantes de la BRDF (les lobes fins). Dans le cas de rendus hors ligne, Veach *et al.* [33] montrent que ces politiques sont complémentaires et propose une méthode permettant de choisir une politique ou une *combinaison* de politiques suivant le calcul à faire.

Notre approche tente de tenir compte de ces observations en restant adaptée au GPU et à une application temps-réel.

2.7 Notre intégration probabiliste

La première idée est d'utiliser la BRDF comme fonction de probabilité de densité. Deux solutions s'offrent à nous, précalculer $P^{-1}(n,\theta,\phi)$ ou opter pour la méthode du rejet.

2.7.1 Précalculer P

Cette solution pose deux problèmes. Le premier est le coût du stockage, en appliquant la décomposition précédente à $p(n,\theta,\phi)$, on obtient deux fonctions $q(n,\theta)$ et $q_\theta(n,\phi)$ donc une fonction $Q^{-1}(n,\theta)$ demandant une texture 2D pour être tabulée et une fonction $Q_\theta^{-1}(n,\phi)$ demandant une texture 3D.

Le second est le coût engendré par la calcul de l'intersection avec les sources de lumières pour chaque échantillon. Un « rayon »ne trouvant pas de lumières sera inutile, mais aussi coûteux qu'un échantillon utile...

Un point intéressant est que les calculs effectués pour choisir un échantillon sont réutilisables pour calculer sa contribution. Le problème est que ces calculs représentent la principale charge de travail.

2.7.2 Une méthode de rejet

Cette méthode semble intéressante car elle permet de placer les échantillons directement sur la source de lumière sans de coûteux précalculs en utilisant la relation [33] :

$$p_{surfaces}(y) = p_{angles}(l)\frac{cos(\theta_y)}{||x-y||^2} \tag{2.34}$$

$p_{surfaces}$ et p_{angles} sont des fonctions de probabilité de densité. Pour rappel, x est une position sur la surface éclairée, y est une position sur un luminaire, l est la direction de x vers y et θ_y est l'angle formé par $-l$ et la normale au point y (direction du luminaire dans notre cas).

Même si elle évite le calcul de l'intersection avec les sources de lumières, elle n'empêche pas les calculs inutiles. En effet cette méthode génère un grand nombre d'échantillons défaussés (la plupart des calculs est réalisée pour générer l'échantillons, mais il ne contribue pas à l'intégration).

2.7.3 Éviter ces calculs inutiles

Nous avons choisi une fonction de probabilité de densité ayant un comportement similaire à celui de la BRDF. Cette fonction a pour but de placer la moitié des échantillons dans la zone de la BRDF représentant la moitié de du volume de la BRDF. Comme pour l'approche biaisée, cette zone est représentable par un cône et l'intersection peut être approximée par une boîte englobante. Nous avons donc une fonction de densité de probabilité de cette forme :

$$p(x,y) = \begin{cases} \frac{0.5}{B} & \text{si (x,y) dans la boîte englobante} \\ \frac{0.5}{C} & \text{sinon} \end{cases} \tag{2.35}$$

L'aire du luminaire $A = B + C$, B est l'aire de la boîte englobante et C le reste. L'utilisation de cette fonction de densité de probabilité permet de réécrire directement l'équation 2.7 sous la forme :

$$L_o(x,v) \approx \sum_i \left(\frac{1}{N} \sum_{j=1}^{N} \frac{f_i(x_p, y_p, x, v)}{p(x_p, y_p)} \right) \tag{2.36}$$

$$= \sum_i \left(\frac{B}{N} \sum_{j=1}^{\frac{N}{2}} f_i(x_j, y_j, x, v) + \frac{C}{N} \sum_{k=1}^{\frac{N}{2}} f_i(x_k, y_k, x, v) \right) \tag{2.37}$$

Avec

$$f_i(x_k, y_k, x, v) = f_r(l^l, v^l) G(x, S_i(x_k, y_k)) V_{2_i}(x, S_i(x_k, y_k)) \tag{2.38}$$

Les échantillons (x_p, y_p) sont distribués suivant p. Les échantillons (x_i, y_i) sont générés uniformément dans la boîte englobante et les échantillons (x_j, y_j) uniformément sur le reste du luminaire. La fonction f_i effectue le calcul d'un échantillon (x_k, y_k) placé sur la $i$ème lumière en $S_i(x_k, y_k)$. l est la direction de x à $S_i(x_k, y_k)$.

La génération des échantillons du reste du luminaire est réalisable de deux manières différentes. La première est une méthode à base de rejet, elle consiste à générer des échantillons uniformément sur la surface du luminaire et à ne conserver que ceux qui sont hors de la boîte englobante. La seconde méthode consiste à découper la zone restante en huit, puis choisir l'une d'entre elles aléatoirement en

fonction de son aire. Une fois la zone sélectionner, il reste à générer un échantillon uniformément dans cette zone. Après test, la première solution s'est révélée plus rapide, nous avons donc choisi de l'utiliser.

Lorsque la source de lumière est proche et que l'exposant est assez élevé, la boîte englobante reste petite. Ceci limite énormément le nombre de rejets (contrairement à la méthode précédente à base de rejet). Lorsque la boîte englobante devient trop grande (aire supérieur à la moitié de l'air du luminaire) il devient judicieux de ne plus considérer p et de se contenter d'échantillonner la source lumineuse.

2.7.4 Discussion

Cette méthode peut paraître séduisante, mais les résultats obtenus sont très bruités même avec un fort échantillonnage. Ce bruit excessif nous empêche d'utiliser peu d'échantillons donc les performances restent faibles (voir figure 2.23). Nous avons envisagé d'effectuer un filtrage « sur l'objet »pour réduire le bruit et l'échantillonnage, mais les premiers résultats ne sont pas convaincants (voir figure 2.24). L'idée du filtrage reste intéressante mais le bruit doit être faible pour que le filtrage ne détruise pas les reflets.

Malgré la réduction de la variance due au fort échantillonnage de la zone dans la boîte englobante qui représente les fortes valeurs de la BRDF, il semble qu'une augmentation est du à la zone restante (voir figure 2.25). Elle réduit ou annule tout le bénéfice apporté par notre fonction de probabilité de densité. Il est certainement possible de jouer sur le gain de variance en répartissant différemment les échantillons, mais les faibles performances de cette approche par rapport à l'approche biaisée ne nous ont pas poussée dans cette direction.

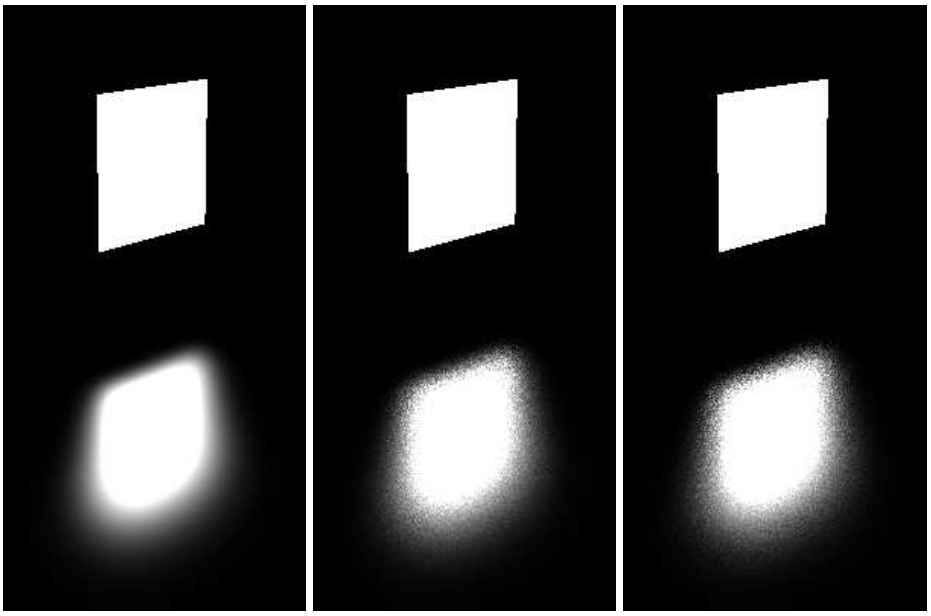

FIG. 2.23 – La scène est composée d'un plan ayant une matière brillante ($e = 200$). À gauche l'image de référence obtenue par la méthode brutale (12 *fps*, pas d'échantillonnage = 0.15). Au centre l'image obtenue avec 256 échantillons distribués uniformément (12 *fps*). À droite l'image obtenue avec 256 échantillons distribués avec notre fonction de probabilité de densité (12 *fps*). Le gain de qualité est presque inexistant.

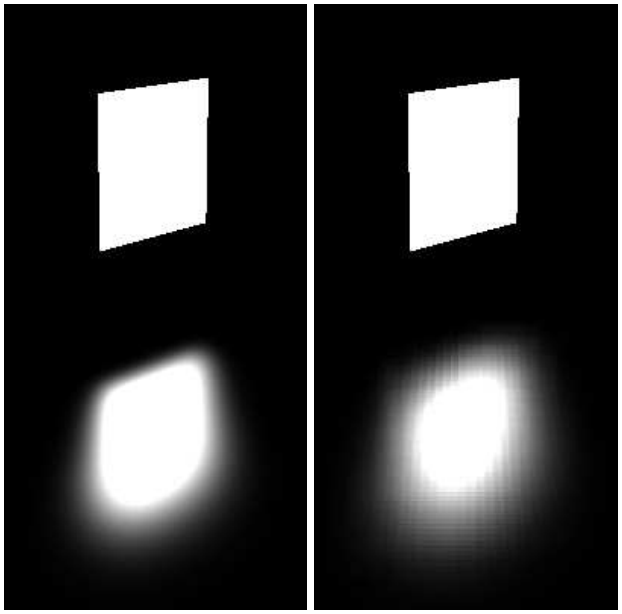

FIG. 2.24 – La scène est identique à celle de la figure 2.23. À gauche l'image de référence obtenue par la méthode brutale (12 *fps*, pas d'échantillonnage = 0.15). À droite l'image obtenue avec 32 échantillons distribués avec notre fonction de probabilité de densité (12 *fps*). Le résultat est filtré sur l'objet en choisissant un filtre adapté au bruit (un filtre plus petit ne réduit pas suffisamment le bruit). Le reflet filtré est plus large et sa partie saturée est réduite.

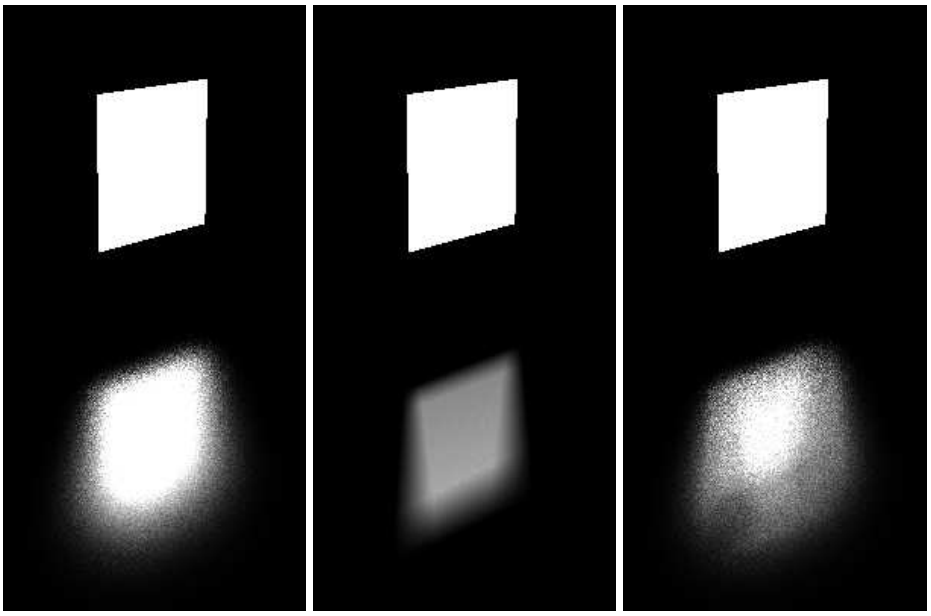

FIG. 2.25 – À gauche l'image obtenue avec la méthode probabiliste. Au centre, la contribution des échantillons placés aléatoirement dans la boîte englobante. À droite, la contribution des échantillons placés hors de la boîte englobante. On remarque que la variance de l'image centrale est très faible et que la variance de l'image de droite reste élevée.

Chapitre 3

Conclusion

Ce dernier chapitre est consacré à l'établissement du bilan de ce stage et au dégagement des perspectives.

3.1 Bilan et Perspectives

Lors de ce stage nous avons proposé une méthode simple et adaptée aux cartes graphiques qui permet d'utiliser des sources surfaciques proche et des matières très brillantes. Nous avons aussi testé et analysé une méthode probabiliste et montré les principaux défauts de cette méthode.

L'approche brutale et l'approche adaptative offrent un rendu de qualité mais leurs performances restent faibles. Par contre, l'approche biaisée offre de très bonnes performances et un rendu fidèle. Contrairement aux autres, la qualité des images générées par l'approche probabiliste n'est pas satisfaisante, le bruit est trop élevé. De plus les performances de cette approche sont faibles. Ces différentes approches permettent toutes d'avoir des scènes totalement dynamiques sans aucun précalcul (un ordre d'idées des performances est donné figure 3.1).

Hormis les ombres douces, l'intérêt des sources surfaciques proches et de voir apparaître leurs formes dans les reflets qu'elles provoquent. Nous avons vu que nous pouvons utiliser une texture pour stocker la forme de luminaires exotiques. Nous pourrions aussi utiliser une texture pour créer des luminaires non parfaits. Un travail intéressant serait d'étudier comment échantillonner ces sources lumineuses pour limiter les effets de crénelage (voir figure 3.2).

Le nombre de fragments traités est important, il serait intéressant de ne considérer que les fragments qui apparaitront non noir lors de l'affichage. Une deuxième idée serait de ne pas calculer l'éclairage pour chaque fragment mais seulement quand la différence d'éclairage est suffisante. L'éclairage d'une grande partie des fragments sera alors interpolé ou extrapolé ce qui réduira le nombre de

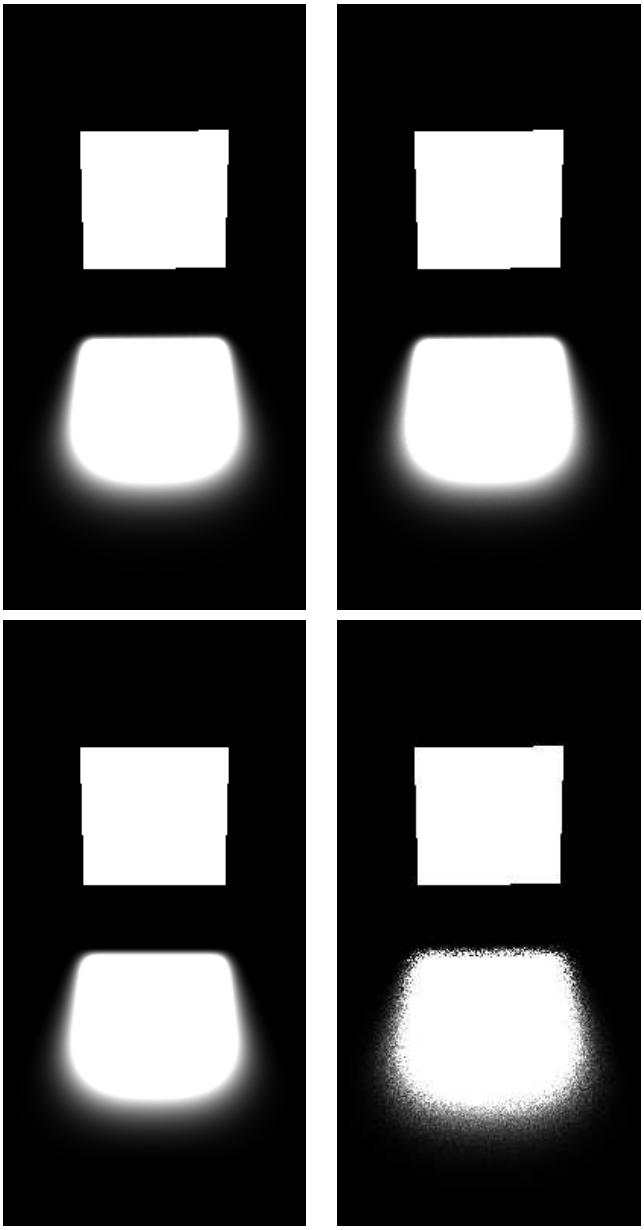

FIG. 3.1 – On trouve la méthode brutale en haut à gauche (12 *fps*), la méthode adaptative en haut à droite (33 *fps*), la méthode biaisée en bas à gauche (220 *fps*), la méthode probabiliste en bas à droite (3 *fps* pour 256 échantillons).

FIG. 3.2 – Un luminaire IVR ! On remarque le crénelage du reflet. Cet artefact est du à l'échantillonage régulier de la source de lumière.

calcul.

Le modèle de Lafortune est une généralisation du modèle de Phong, il serait intéressant d'étendre le travail effectué pour pouvoir bénéficier d'une plus grande palette de matières.

3.2 Remerciements

Je tiens à remercier Nicolas Holzschuch pour son encadrement et ses nombreux conseils, Eric Bruneton pour avoir accepté d'être mon rapporteur, Elmar Eisemann pour ses avis éclairés sur la programmation des GPU et son générateur de valeurs aléatoires, David Roger pour son aide sur les coniques et la relecture de ce rapport, Xavier Décoret et Cyril Soler pour leurs coups de mains lorsque ma machine refusait de fonctionner, Pierre-Edouard Landes et Emmanuel Turquin pour leurs relectures et conseils, ainsi que tout le reste de l'équipe ARTIS pour la bonne ambiance qu'il y règne.

Bibliographie

[1] T. Annen, J. Kautz, F. Durand, and H.-P. Seidel. Spherical harmonic gradients. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Sketches*, page 110, New York, NY, USA, 2004. ACM Press.

[2] J. Arvo. The irradiance jacobian for partially occluded polyhedral sources. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 343–350, New York, NY, USA, 1994. ACM Press.

[3] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77 : Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, New York, NY, USA, 1977. ACM Press.

[4] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. volume 19, pages 542–547, New York, NY, USA, 1976. ACM Press.

[5] B. Cabral, M. Olano, and P. Nemec. Reflection space image based rendering. In A. Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 165–170, Los Angeles, 1999. Addison Wesley Longman.

[6] M. F. Cohen, J. Wallace, and P. Hanrahan. *Radiosity and realistic image synthesis*. Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[7] C. R. H. Gene S. Miller. Illumination and reflection maps : Simulated objects in simulated and real environments. In *SIGGRAPH '84 Advanced Computer Graphics Animation seminar notes*, 1984.

[8] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM Press.

[9] W. Heidrich and H.-P. Seidel. Realistic, hardware-accelerated shading and lighting. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 171–178, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[10] J. T. Kajiya. The rendering equation. In *SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, New York, NY, USA, 1986. ACM Press.

[11] N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. In *Neural Computation*, 1997.

[12] J. Kautz and M. D. McCool. Interactive rendering with arbitrary brdfs using separable approximations. In *Proceedings of the Eurographics Workshop on Rendering Techniques 1999*, pages 185–196, 1999.

[13] J. Kautz and M. D. McCool. Approximation of glossy reflexion with prefiltered environment maps. In *Proceedings Graphics Interface 2000*, pages 119–126, 2000.

[14] J. Kautz, P.-P. Sloan, and J. Snyder. Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering*, pages 291–296, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[15] J. Kautz, P.-P. Vàzquez, W. Heidrich, and H.-P. Seidel. Unified approach to prefiltered environment maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 185–196, London, UK, 2000. Springer-Verlag.

[16] A. W. Kristensen, T. Akenine-Möller, and H. W. Jensen. Precomputed local radiance transfer for real-time lighting design. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 1208–1215, New York, NY, USA, 2005. ACM Press.

[17] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[18] H. P. A. Lensch, M. Goesele, Y.-Y. Chuang, T. Hawkins, S. Marschner, W. Matusik, and G. Mueller. Realistic materials in computer graphics. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Courses*, page 1, New York, NY, USA, 2005. ACM Press.

[19] D. K. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *HWWS '02 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 79–88, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[20] M. D. McCool, J. Ang, and A. Ahmad. Homomorphic factorization of brdfs for high-performance rendering. In *SIGGRAPH '01 : Proceedings of the*

28th annual conference on Computer graphics and interactive techniques, pages 171–178, New York, NY, USA, 2001. ACM Press.

[21] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 376–381, New York, NY, USA, 2003. ACM Press.

[22] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers*, pages 477–487, New York, NY, USA, 2004. ACM Press.

[23] A. Ngan, F. Durand, and W. Matusik. Experimental validation of analytical brdf models. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Sketches*, page 90, New York, NY, USA, 2004. ACM Press.

[24] B. T. Phong. Illumination for computer generated pictures. volume 18, pages 311–317, New York, NY, USA, 1975. ACM Press.

[25] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, New York, NY, USA, 2001. ACM Press.

[26] S. Rusinkiewicz. A new change of variables for efficient brdf representation. In G. Drettakis and N. Max, editors, *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pages 11–22, New York, NY, 1998. Springer Wien.

[27] P. S. Shirley. *Physically based lighting calculations for computer graphics*. PhD thesis, Champaign, IL, USA, 1991.

[28] P.-P. Sloan. Normal mapping for precomputed radiance transfer. In *I3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 23–26, New York, NY, USA, 2006. ACM Press.

[29] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder. Clustered principal components for precomputed radiance transfer. In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 382–391, New York, NY, USA, 2003. ACM Press.

[30] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, New York, NY, USA, 2002. ACM Press.

[31] P.-P. Sloan, B. Luna, and J. Snyder. Local, deformable precomputed radiance transfer. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 1216–1224, New York, NY, USA, 2005. ACM Press.

[32] L. Szirmay-Kalos, B. Aszódi, I. Lazányi, and M. Premecz. Approximate ray-tracing on the gpu with distance impostors. *Computer Graphics Forum (Proc. Eurographics 2005)*, 24(3), 2005.

[33] E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, New York, NY, USA, 1995. ACM Press.

[34] J. Wang, K. Xu, K. Zhou, S. Lin, S. Hu, and B. Guo. Spherical harmonics scaling. *Vis. Comput.*, 22(9) :713–720, 2006.