

Lancer de photons multi-passes et écrasement de photons pour le rendu optronique.

THESE

présentée et soutenue publiquement le 20 janvier 2006

pour l'obtention du

Doctorat de l'Université Paul Sabatier

(Spécialité Informatique)

par

Antoine Boudet

Composition du jury

Président : René Caubet

Rapporteurs : Kadi Bouatouch
Michel Mériaux

Examineurs : Mathias Paulin
Paul Pitot

Résumé

LANCER DE PHOTONS MULTI-PASSES ET ÉCRASEMENT DE PHOTONS POUR LE RENDU OPTRONIQUE

Mots-clés: Synthèse d'image, optronique, infrarouge, diffusion, illumination globale, lancer de photons, écrasement de photons, milieu participant, estimation de densité

La simulation de l'éclairage par illumination globale a fait l'objet de nombreuses recherches et applications au cours des dernières années. Tout d'abord utilisée dans le domaine visible, la simulation est aujourd'hui de plus en plus appliquée au rendu infrarouge. On appelle *optronique* l'union de ces deux domaines. Le problème principal des méthodes d'illumination globale actuelles provient de la difficulté à traiter le phénomène de diffusion de la lumière, aussi bien dans le cas des surfaces que des milieux participants. Ces méthodes offrent des résultats satisfaisants dans le cas de scènes simples, mais les performances s'effondrent lorsque la complexité augmente. Dans la première partie de cette thèse, nous exposons la nécessité de la prise en compte des phénomènes de diffusion pour la simulation optronique. Dans la deuxième partie nous posons les équations qui unifient les différentes méthodes de synthèse d'image, c'est-à-dire l'équation du rendu et l'équation volumique du transfert radiatif. L'état de l'art des méthodes d'illumination globale présenté dans la troisième partie montre qu'à l'heure actuelle la méthode des cartes de photons est celle qui offre le meilleur compromis performance/qualité. Néanmoins, la qualité des résultats obtenus grâce à cette méthode est dépendante du nombre de photons qui peuvent être stockés et donc de la quantité de mémoire disponible. Dans la quatrième partie de la thèse, nous proposons une évolution de la méthode, *le lancer de photons multi-passes*, qui permet de lever cette dépendance mémoire, et ainsi d'obtenir une très grande qualité sans pour autant utiliser une configuration matérielle onéreuse. Un autre problème de la méthode des cartes de photons est le temps de calcul important nécessaire lors du rendu de milieux participants. Dans la cinquième et dernière partie de cette thèse, nous proposons une méthode, *l'écrasement de photons volumique*, qui prend avantage de l'estimation de densité pour reconstruire efficacement la luminance volumique à partir de la carte de photons. Notre idée est d'isoler le calcul de la

diffusion et d'utiliser une approche duale de l'estimation de densité pour l'optimiser car il constitue la partie coûteuse du calcul. Bien que les temps de rendu obtenus par notre méthode sont meilleurs que ceux obtenus en utilisant la méthode des cartes de photons pour la même qualité, nous proposons aussi une optimisation de la méthode utilisant les nouvelles capacités des cartes graphiques.

Abstract

MULTIPASS PHOTON MAPPING AND PHOTON SPLATTING FOR OPTRONIC RENDERING

Keywords: Image synthesis, optronic, infrared, scattering, global illumination, photon mapping, photon splatting, participating media, density estimation

Much research have been done on global illumination simulation. Firstly used in the visible spectrum domain, today, simulation is more and more applied to infrared rendering. The union of these two domains is called *optronic*. The main problem of the current global illumination methods comes from the complexity of the light scattering phenomena, as well for surfaces as for participating media. These methods offer satisfactory results for simple scenes, but performances crash when complexity raises. In the first part of this thesis, we expose the necessity to take scattering phenomena into account for optronic simulation. In the second part, we pose the equations that unify all global illumination methods, i.e. the rendering equation and the volume radiative transfer equation. The state of the art presented in the third part shows that the Photon Mapping method is, at this moment, the one that offers the better compromise between performance and quality. Nevertheless, the quality of the results obtained with this method depends on the number of photons that can be stored, and then on the available memory. In the fourth part, we propose an evolution of the method, called *Multipass Photon Mapping*, which permits to get rid of this memory dependency, and hence, to achieve a great accuracy without using a costly hardware configuration. Another problem inherent to Photon Mapping, is the enormous rendering time needed for participating media rendering. In the fifth and last part of this thesis, we propose a method, called *Volume Photon Splatting*, which takes advantage of density estimation to efficiently reconstruct volume radiance from the photon map. Our idea is to separate the computation of emission, absorption and out-scattering from the computation of in-scattering. Then we use a dual approach of density estimation to optimize this last part as it is the most computational expensive. Our method extends Photon Splatting, which optimizes the computation time of Photon Mapping for surface rendering, to participating media, and then considerably reduce participating media rendering times. Even though our method is faster than Photon Mapping for equal quality, we also propose a GPU based optimization of our algorithm.

Remerciements

Merci à tous ! Et voilà. Ca, c'est fait... Comment ? Ca suffit pas ? Parce que vous voyez, c'est compliqué les remerciements. C'est comme le foie gras, il faut pas en manger trop sinon ça perd de sa saveur, mais il en faut assez sinon gare à la frustration ! En plus, on ne sait pas s'il faut être formel ou non, et il ne faut oublier personne... Allez, je me lance.

Je voudrais tout d'abord remercier mes rapporteurs M. le professeur Kadi Bouatouch, de l'université de Rennes, et M. le professeur Michel Mériaux, de l'université de Poitiers, qui m'ont jugé digne de devenir Docteur des ordinateurs. Je m'excuse encore auprès d'eux pour avoir planifié mon pot de thèse à 17 heures et non juste après la soutenance, ne sachant pas qu'ils avaient leur avion en début d'après-midi.

Je remercie M. le professeur René Caubet, Directeur de Recherche du Laboratoire de Synthèse d'images, de m'avoir permis d'entrer dans son équipe, et pour m'avoir fait découvrir l'OpenGL ce qui a fortement contribué à me mener cahin caha vers mes sujets de recherche.

Je remercie M. Mathias Paulin, mon directeur de thèse, pour les innombrables fois où j'ai posé des questions farfelues, ou compliquées, auxquelles il a toujours répondu dans la bonne humeur. Je le remercie également pour m'avoir guidé, aidé et soutenu tout au long de ces années. Et merci d'être toujours rentré par les coteaux.

Je remercie M. Jean Latger, Président Directeur Général d'Oktal Synthetic Environment, de m'avoir embauché dans son entreprise et d'avoir permis de mettre en place la structure collaborative ayant mené à l'achèvement de mes travaux.

Je remercie Le Maître, M. Paul Pitot, Directeur Recherche et Développement, de m'avoir encadré au sein d'Oktal Synthetic Environment, guidé, aidé, et d'avoir permis la naissance des nouvelles méthodes présentées ici. Merci aussi pour tout tes soins, Doc !

Je remercie toute l'équipe de Synthèse d'images pour la très bonne ambiance qui a régné tout au long des trois ans et demi : Luc, chaud comme la braise, Fabien, le pc est à moi, Yann, et le roi de coeur, Olivier, le plus sympa, Ze, toujours la tête dans la lune, Piqueuh, à qui je tire mon chapeau, Anca, et son pantalon blanc, Gaël, et son vélo, Mathieu, et mes ronflements, Loïc et ses bons conseils. Je suis convaincu que les innombrables parties de tarot, d'échecs, les pots et les barbecues ont fortement contribué à la réussite de mon doctorat.

Je remercie mes collègues et amis qui m'ont accepté au sein d'Oktal-SE et avec qui la rigolade était autant au rendez-vous que le travail : Vincent, mon ours du Gers préféré, Lionel, toujours

là, Pierre, le canyoning, Christophe, pour tous les apéros et les jeux avec Véro, Carole, les soirées pizzas, Myriam, la chute libre, Myriam, le roller, Florent, et les chameaux sauvages, Stéphane, dit Grumly, Lalaina, et la pause thé, Carlos, et son tact légendaire, Arnold, simplement métal, Nicolas, my nose, David, ainsi que son sceptre et Thierry, pour toutes ces pauses cafés.

Je remercie aussi mes amis qui étaient toujours là pour faire la fête mais également pour me remonter le moral dans les moments difficiles : Loïc, Mathieu, Nanou, Nash, Cyril, Thomas, Stéphane, Titelo, Ludo, Marjorie, Ben, Emeuh, Tut et Matthieu sans l'imprudence de qui cette thèse n'aurait jamais été publiée. WoW et une thèse en même temps c'est pas toujours facile. Je n'ai pas besoin de rentrer dans le détail, mais sachez que grâce à vous ces années resteront inoubliables pour moi.

Je remercie tous ceux que j'aurai par mégarde oublié ici et qui auraient mériter d'y figurer.

Enfin, et surtout, je remercie ma famille qui m'a toujours soutenu et m'a permis d'arriver jusqu'ici. Merci Bro ! Merci tontons, tatas, zins et zines ! Merci Pépé et Muttie. Merci Maman, Merci Papa. Tous les ans, je voudrais que ça r'commence ! You kaïdi aïdi aïda. Euh... En fait, non... C'est bien comme ça.

A tous, un grand MERCI !

A Samuel, mon poto...

Table des matières

Table des figures	xiii
Introduction	1
Chapitre 1 Problématiques	5
1.1 Simulation optronique	6
1.2 Milieux participants	9
Chapitre 2 Théorie du transfert radiatif	13
2.1 Introduction	13
2.2 Equation du rendu	13
2.3 Milieux participants	15
2.3.1 Phénomènes élémentaires	15
2.3.2 Absorption	15
2.3.3 Emission	16
2.3.4 Diffusion	17
2.4 Equation volumique du transfert radiatif	20
2.5 Conclusion	20
Chapitre 3 Calcul de l'illumination globale	23
3.1 Introduction	23
3.2 Surfaces	24
3.2.1 Méthodes de Radiosité	24
3.2.2 Méthodes de Lancer de rayons	29
3.2.3 Illumination globale par estimation de densité	37
3.3 Milieux participants	45
3.3.1 Méthodes déterministes	46
3.3.2 Méthodes stochastiques	50
3.4 Conclusion	53

Chapitre 4 Lancer de photons multi-passes	55
4.1 Introduction	55
4.2 Détails de la méthode des cartes de photons	55
4.2.1 Description d'un photon	56
4.2.2 Emission des photons	56
4.2.3 Propagation des photons	57
4.2.4 Stockage des photons	58
4.2.5 Reconstruction de la luminance	58
4.3 Lancer de photons spectral	60
4.3.1 Modèle spectral	61
4.3.2 Choix du modèle spectral	63
4.3.3 Photons spectraux	64
4.3.4 Réflexions spectrales	65
4.3.5 Estimation spectrale de la luminance	66
4.4 Approche multi-passes	68
4.4.1 Problèmes dus à l'estimation de densité spectrale	68
4.4.2 Méthodes de cache existantes	70
4.4.3 Principe de base de l'approche multi-passes.	71
4.4.4 Reconstruction de l'image finale	72
4.4.5 Parallélisme	75
4.4.6 Algorithme	75
4.5 Résultats	75
4.5.1 Tube spéculaire	76
4.5.2 Salle de conférence du Soda Hall	77
4.5.3 Discussion	79
4.6 Conclusion et perspectives	80
Chapitre 5 Ecrasement de photons pour les milieux participants	83
5.1 Introduction	83
5.2 Discussion de la méthode des cartes de photons pour les milieux participants	83
5.3 Formulation itérative de l'équation volumique du transfert radiatif	85
5.4 Description du milieu	87
5.5 Simulation de l'émission, de l'absorption et de la diffusion sortante	88
5.6 Simulation de la diffusion entrante	89
5.6.1 Description de l'écrasement de photons	89
5.6.2 Rendu volumique	91
5.6.3 Principe de l'écrasement de photons volumique	93

5.7	Algorithme de l'écrasement de photons	101
5.8	Optimisation GPU	101
5.8.1	Notions d'implémentation GPU	102
5.8.2	Principe	102
5.8.3	Précision	103
5.8.4	Résultats	104
5.9	Conclusion et perspectives	106
Conclusion		109
Annexes		111
Annexe A Données radiatives		111
A.1	Flux	111
A.2	Eclairement	111
A.3	Luminance	111
A.4	Fonction de réflectance bidirectionnelle	111
A.5	Fonction de phase	112
A.6	Loi du corps noir	113
Annexe B Fonction de densité de probabilité		115
B.1	Définition	115
B.2	Echantillonnage par importance	116
B.2.1	Principe	116
B.2.2	Mise en oeuvre	116
Annexe C Réalisations industrielles		119
C.1	SPECRAY	119
C.1.1	Description	119
C.1.2	Apports	119
C.2	COMPIR : Visualisation infrarouge temps réel	120
C.2.1	Description	120
C.2.2	Apports	121
C.3	RDVIS : Modèle de ciel temps réel	122
C.3.1	Modèles de ciel	122
C.3.2	Caractérisation des sources lumineuses	123
Bibliographie		129

Table des figures

1	Première image photo-réaliste de synthèse d'image.	3
1.1	Cycle de vie en V d'un système complexe.	5
1.2	Luminance du corps noir par la loi de Planck	9
1.3	Spectre d'absorption du CO_2 en infrarouge.	11
2.1	Géométrie associée à la formulation directionnelle de l'EQR.	14
2.2	Géométrie associée à la formulation surfacique de l'EQR.	15
2.3	Géométrie liée à l'équation volumique du transfert radiatif	16
2.4	Phénomènes modifiant la propagation de la lumière dans un milieu participant .	17
2.5	Fonction de phase de Heyney-Greenstein	18
3.1	Principe du lancer de rayons	30
3.2	Principe du lancer de rayons stochastique.	33
3.3	Principe du tracer de chemins.	34
3.4	Principe du lancer de particules.	36
3.5	Principe du tracer de chemins bidirectionnel.	36
3.6	Estimation de densité par noyau.	38
3.7	Estimation de densité	44
4.1	Représentation d'un photon	56
4.2	Principe de l'émission des photons.	57
4.3	Exemple de kd-tree en deux dimensions	59
4.4	Visualisation directe de la carte de photons	60
4.5	Visualisation de la carte de photons au deuxième rebond	60
4.6	Réorganisation des coefficients d'absorption par la méthode des k -distributions. .	64

4.7	Approximation du spectre par un modèle de bandes grises	64
4.8	Rendu infrarouge d'un tube émissif et spéculaire	68
4.9	Problème d'estimation de densité pour les matériaux spéculaires.	69
4.10	Principe du lancer de photons multi-passes	72
4.11	Illustration de l'équivalence entre un rendu avec une et trente passes	73
4.12	Principe du parallélisme du lancer de photons multi-passes.	76
4.13	Evolution du calcul multi-passes	79
4.14	Comparaison entre le lancer de photons spectral simple et multi-passes	80
4.15	Rendu de la salle de conférence du Soda Hall	81
5.1	Illustration du bruit poivre et sel dû au manque d'échantillons.	84
5.2	Redondance des recherches par la méthode des cartes de photons	85
5.3	Structure de données utilisée pour représenter le milieu participant.	88
5.4	Méthode du ray marching	89
5.5	Principe de l'écrasement de photons	90
5.6	Evolution de l'écrasement de photons	91
5.7	Géométrie associée à l'écrasement de photons.	96
5.8	Profil 1D des différentes fonctions noyaux et les textures intégrées associées . . .	98
5.9	Principe de l'écrasement de photons volumique	100
5.10	Pipeline OpenGL classique.	102
5.11	Pipeline OpenGL programmable.	103
5.12	Rendu d'un nuage anisotropique	104
5.13	Rendu d'une parabole placée dans nuage	106
A.1	Géométrie locale à la surface associée à la formulation de la FDRB.	112
A.2	Luminance du corps noir donnée par la loi de Planck	113
B.1	Fonction de densité de probabilité gaussienne (ou loi normale).	116
B.2	Mécanisme d'échantillonnage selon un loi de probabilité inversible.	117
B.3	Mécanisme d'échantillonnage par rejet.	118
C.1	Exemple d'images générées par SPECRAY	120
C.2	Rendu infrarouge temps réel : COMPIR	121
C.3	Représentation de modèles de ciel	122

C.4	Choix de la couleur de l'éclairage en fonction de l'angle du soleil avec la terre . .	124
C.5	Méthode de plaquage des textures de nuages.	126
C.6	Evolution du modèle de ciel et d'éclairage.	127

Introduction

La synthèse d'image a pour but de représenter le plus fidèlement possible le monde réel. La figure 1 présente la première image photo-réaliste calculée avec une méthode type radiativité. On était alors loin de générer des images que l'on peut confondre avec la réalité. Avec l'apparition de la synthèse d'image photo-réaliste, un effort important a été déployé pour augmenter le réalisme des images générées. Nous verrons dans le chapitre 3 que de nombreuses méthodes ont été proposées en ce sens. Cependant, un grand nombre de ces méthodes dites d'illumination globale, donnant des résultats très satisfaisants pour des scènes simples, sont mises en échec lorsque la complexité des scènes augmente. Ceci parce que les ordinateurs actuels, bien que puissants, possèdent des limites en terme de calcul et de taille mémoire. Certaines méthodes, et en particulier celle des cartes de photons, offrent néanmoins un compromis intéressant entre temps et qualité de rendu.

Cette thèse possède deux axes de recherche distincts mais regroupés au sein d'une même problématique, que nous décrivons en détail au chapitre 1. Dans le contexte de la simulation optronique, qui regroupe les domaines du visible et de l'infrarouge, notre recherche porte sur la simulation de la diffusion. En effet, un grand nombre d'applications utilisent aujourd'hui la simulation infrarouge tant dans le domaine civil que militaire. Or il n'existe que peu de méthodes de simulation infrarouge, et la majorité d'entre elles font des hypothèses simplificatrices importantes et notamment ignorent ce phénomène. Notre but a été de pallier ce manque. Notre premier axe de recherche a porté sur le calcul de ce phénomène dans le cas de scènes composées uniquement de surfaces et notamment de résoudre les problèmes de mémoires inhérents à la complexité du phénomène. Le deuxième axe concerne la minimisation des temps de calcul nécessaires à la simulation de ce même phénomène, mais, cette fois, dans le cas des milieux participants (nuages, gaz, fumées...) où la complexité est encore plus importante.

Après avoir exposé en détail la problématique dans laquelle ont été menés nos travaux de

recherche, nous décrivons les modèles physiques qui régissent les transferts de lumière dans une scène 3D (chapitre 2). Le but des méthodes de synthèse d'image est de résoudre ces modèles pour tous les éléments de la scène. Néanmoins, la complexité est telle que toutes les méthodes font des hypothèses simplificatrices afin de rendre les calculs réalisables dans un temps fini. Certaines de ces hypothèses sont spécifiques à la simulation du domaine visible car elles se basent sur les propriétés du système visuel humain. Or, pour la simulation infrarouge, l'observateur n'est plus l'oeil humain, mais un capteur qui possède ses caractéristiques propres. Ces hypothèses, n'étant plus valides, les méthodes doivent être adaptées pour s'adapter au rendu infrarouge. Suite à l'inventaire des méthodes existantes (chapitre 3), la méthode des cartes de photons apparaît comme le meilleur choix. Nous présentons les modifications que nous avons apportées afin de l'adapter au rendu spectral pour l'infrarouge dans le chapitre 4. Cette adaptation a donné naissance à la méthode du *lancer de photons multi-passes* nécessaire pour résoudre les problèmes de mémoire rencontrés.

La représentation des milieux participants est essentielle pour de nombreuses applications infrarouges comme, par exemple, la détection de points chauds lors de la conception de moteurs dans le domaine de l'aéronautique. La méthode des cartes de photons permet de prendre en compte ce type de milieux. Néanmoins, l'approche utilisée est très gourmande en temps de calcul, et sa version spectrale encore plus. Dans le chapitre 5, nous proposons une méthode basée sur une approche duale de l'estimation de densité qui minimise le temps de rendu nécessaire pour générer une image.

Les algorithmes que nous proposons dans cette thèse ont été industrialisés dans le logiciel SPECRAY de la société OKTAL Synthetic Environment présenté dans l'annexe C.1. Ils ont ainsi été validés et sont activement utilisés par les clients de la société.

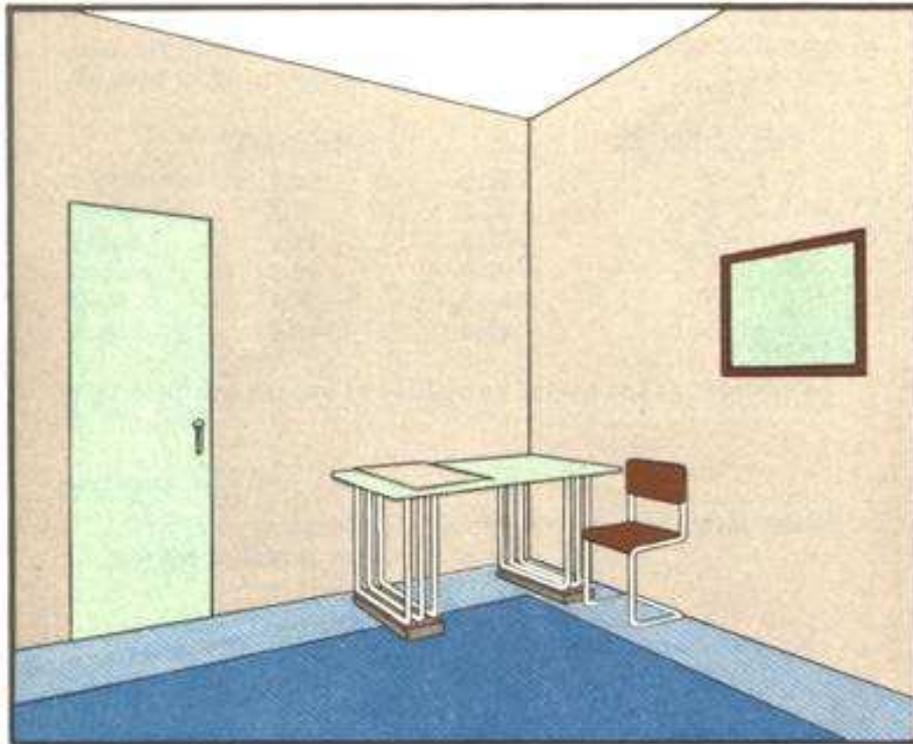


Figure 1. Première image photo-réaliste de synthèse d'image. Parry Moon et Domina Spencer (MIT), *Lighting Design*, 1948. Bien avant l'introduction en synthèse d'image de la méthode de radiosité par Goral en 1984, des méthodes similaires étaient utilisées dans l'industrie pour le calcul d'illumination (i.e. Yamauti 1926, Higgle 1934). Mais aucune image n'était générée. Moon et Spencer calculèrent à la main les luminances de chaque élément de surface de la scène ci-dessus. Ils découpèrent des morceaux de papiers de la bonne couleur et les assemblèrent pour former leurs images. Les images étaient ensuite photographiées avant leur présentation.

Chapitre 1

Problématiques

Nos travaux s'inscrivent dans le domaine de la simulation. Dans ce cadre, on peut distinguer deux catégories : la simulation d'entraînement et la simulation d'étude. Dans le premier cas, la simulation interagit avec un utilisateur humain, dans un but de formation par exemple (e.g. simulateur de conduite de train). Ce type de simulation est contraint par un besoin d'interactivité avec l'utilisateur, les technologies mises à son service sont donc nécessairement temps-réel, le réalisme des résultats étant secondaire. Quant à elle, la simulation d'étude permet d'optimiser les développements de systèmes complexes (e.g. avion, missile...). Dans ce cas, elle peut intervenir à toutes les étapes du cycle de vie d'un tel système (cf. figure 1.1). En phase de spécification, elle

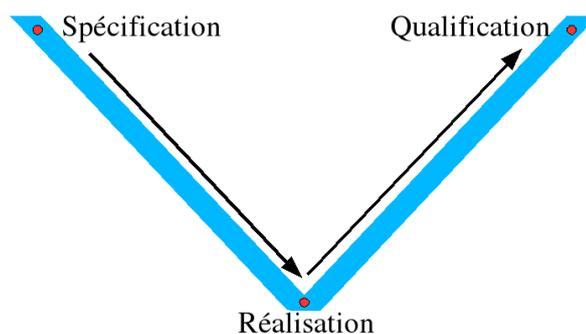


Figure 1.1. Cycle de vie en V d'un système complexe.

offre une aide pour choisir la technologie à utiliser. Par exemple, elle permet de choisir les capteurs utilisés par le système en fonction de leurs propriétés spectrales. Lors de la réalisation, elle permet de tester les algorithmes périphériques (e.g. traitement d'image). Lors de la qualification du système, elle est utilisée pour valider le système avant les tests réels. D'une manière générale, la simulation réduit les coûts liés au cycle de vie du système. En effet, son avantage principal est

de pouvoir balayer l'ensemble des cas du réel, et surtout ceux qui sont difficiles voire impossibles à mettre en place. Elle permet notamment d'effectuer des études paramétriques, c'est à dire de fixer tous les paramètres de la simulation sauf un et de faire varier ce paramètre. Par exemple, en infrarouge, rejouer la simulation plusieurs fois en faisant varier la température mais en gardant les mêmes conditions atmosphériques. La simulation d'étude est effectuée grâce à deux types d'outils :

- ceux qui proposent une simulation temps réel dégradée, qui permet de générer des images en entrée de systèmes complexes pour observer ses réactions suivant des conditions particulières,
- ceux qui proposent une simulation physique et précise qui permet de valider les premiers. Dans ce cas, les technologies utilisées sont qualifiées de hors temps réel car les calculs nécessaires pour obtenir la précision suffisante sont très complexes et très longs.

Nos travaux de recherche se positionnent dans ce dernier cas. Les méthodes de rendu que nous présentons sont destinées à accroître le réalisme de la simulation. En effet, la précision obtenue dépend à la fois des données sur lesquelles se base la simulation et sur les algorithmes de rendu utilisés pour les exploiter. Cependant, à l'heure actuelle, il est très coûteux d'obtenir des données mesurées dans le monde réel, les mécanismes de mesures étant complexes et difficiles à mettre en place. La simulation a l'avantage de pouvoir extrapoler ces données. On peut ainsi, à partir de peu de données, réaliser des simulations sur des scènes complexes, ces simulations étant ensuite recalées par rapport à la réalité par de nouvelles campagnes de mesures. Le réalisme de la simulation peut également être augmenté en proposant de meilleures méthodes d'éclairage.

1.1 Simulation optronique

C'est sur ce dernier point que nous avons travaillé, et notamment sur le calcul de la diffusion, qui est le phénomène lumineux le plus complexe à évaluer car chaque point de la scène est dépendant de tous les autres. Au démarrage de notre recherche, l'outil de simulation optronique (qui couvre les spectres visible et infrarouge), SPECRAY, utilisait un modèle optimisé de lancer de rayons spectral. Dans ce modèle initial, ce phénomène n'était pas pris en compte. Celui-ci s'avère cependant nécessaire pour obtenir une précision suffisante dans certains cas de simulations (brouillards, obscurcissants...). L'omission de la diffusion ne porte pas à conséquence pour certaines utilisations comme, par exemple, pour l'observation de terrains. Mais dans le cas

de l'étude des phénomènes thermiques dans des moteurs, leur importance est cruciale. En effet, les phénomènes thermiques peuvent très bien y générer des points chauds, et ceux-ci sont indétectables si l'on ne considère pas la diffusion et les focalisations de lumière (aussi appelées *caustiques*).

On appelle simulation optronique, la simulation des ondes électromagnétiques dont la longueur d'onde est de l'ordre du micron. Elle regroupe les longueurs d'ondes du domaine visible ($\lambda \in [4.10^{-7}, 8.10^{-7}]$ mètres) et la simulation infrarouge pour laquelle on trouve couramment une découpe en trois bandes (cf. figure 1.2) :

- Bande I : $\lambda \in [1.10^{-6}, 2.10^{-6}]$ mètres.
- Bande II : $\lambda \in [3.10^{-6}, 5.10^{-6}]$ mètres.
- Bande III : $\lambda \in [8.10^{-6}, 12.10^{-6}]$ mètres.

Cette découpe a l'avantage de couvrir l'ensemble de la variété de phénomènes du domaine infrarouge. En bande I, comme en visible, le phénomène de diffusion est prépondérant sur le phénomène d'émission propre. Au contraire, en bande III, la diffusion est quasiment absente et l'émission est très forte. Finalement, en bande II, les deux ont une importance quasi-équivalente.

Nous souhaitons uniformiser les traitements effectués entre les domaines visible et infrarouge. Or, peu de travaux ont été effectués dans le cadre de la synthèse d'image spécifiquement pour ce dernier. Il existe néanmoins un grand nombre de méthodes de simulation du visible, celles-ci offrant une base solide pour l'extension à l'infrarouge.

Contraintes dues à l'infrarouge La puissance et la capacité de stockage des ordinateurs ne sont pas infinies, bien qu'elles aient connu une augmentation exponentielle depuis les débuts de l'informatique. Les modèles physiques à simuler étant, la plupart du temps, très complexes, il est souvent nécessaire de les simplifier. Des hypothèses sont alors faites afin de réduire leur complexité.

La simplification incontournable en informatique concerne la continuité du signal. Que ce soit pour le son, la lumière ou encore les ondes radars, les modèles physiques expriment des fonctions continues. Or, les ordinateurs ne traitent que des données discrètes. Des descriptions discrètes des signaux continus sont donc toujours utilisées.

Nous verrons dans la section 3.2 que les méthodes utilisées pour le rendu dans le spectre visible font des hypothèses portant sur :

- le modèle d'éclairage,

- le modèle de matériaux,
- la complexité géométrique de la scène,
- la perception de la lumière par l’œil.

Sous le couvert de ces hypothèses, certaines de ces méthodes offrent des résultats très satisfaisants, tant en termes de qualité de rendu que de rapidité de la simulation. Malheureusement, certaines des hypothèses choisies sont spécifiques à la simulation visible et ne sont plus applicables dans le cadre infrarouge.

La première, et certainement la plus importante, est que, pour les longueurs d’onde du visible, seuls les matériaux très chauds sont émissifs et peuvent donc être considérés comme des sources lumineuses. Nous verrons que la plupart des méthodes de synthèse d’images peuvent être mises en échec lorsque cette affirmation n’est pas vérifiée. Ces méthodes identifient explicitement les sources lumineuses. Cela leur permet de leur appliquer un traitement particulier. En général, cela consiste à échantillonner l’ensemble de ces sources lors du calcul de l’éclairage pour chaque point (ou élément de surface) de la scène. La complexité calculatoire dépend donc fortement du nombre de sources à traiter, et les temps de calcul deviennent rapidement prohibitifs lorsque les sources deviennent plus nombreuses. Or, dans le spectre infrarouge (notamment en bande II et III), tous les objets peuvent être considérés comme des sources lumineuses étant donné que l’émissivité est très sensible à la température pour ces longueurs d’ondes. La luminance émise par un matériau est donnée par la loi du corps noir (ou loi de Planck). Cette loi donne la luminance émise en fonction de la longueur d’onde et de la température. La figure 1.2 présente cette loi. Par exemple, une table en bois à 300 degrés Kelvins (27°C) rayonne principalement en infrarouge bande III.

Ensuite, la complexité spectrale entre en jeu. Les méthodes utilisées pour le rendu visible se contentent la plupart du temps du modèle RVB (Rouge, Vert, Bleu) pour la représentation du spectre visible. Cependant, dans l’infrarouge, certaines longueurs d’onde sont susceptibles d’être absorbées par les matériaux. On parle dans ce cas de raies d’absorption. Une modélisation du spectre par bandes grises (généralisation du modèle RVB pour les autres longueurs d’ondes où l’on suppose que la valeur du spectre est constante sur la bande) n’est alors pas satisfaisante. Dans notre domaine d’application, il est donc nécessaire d’utiliser un modèle spectral qui puisse s’adapter aux variations fines et abruptes du spectre lumineux.

A l’issue de l’état de l’art du chapitre 3, nous verrons que la méthode des cartes de photons offre une base solide pour le rendu des effets d’illumination globale. Nous montrerons dans le

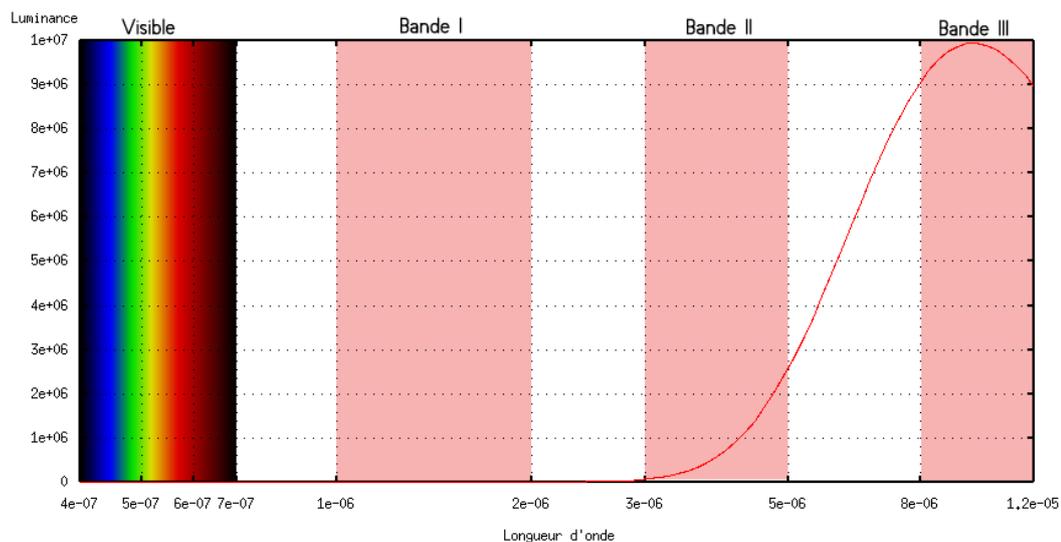


Figure 1.2. Luminance du corps noir donnée par la loi de Planck pour une température de 300K. Un tel corps ne rayonne pas dans le domaine visible mais principalement en bande II et III.

chapitre 4 comment nous l’avons adapté au domaine spectral pour le rendu infrarouge. Nous expliquerons les problèmes de consommation mémoire, dus à cette adaptation, que nous avons résolus par une approche multi-passes. L’approche que nous proposons ne nécessite pas de configuration particulière bien qu’elle puisse en tirer avantage. En effet, certaines méthodes utilisent le parallélisme afin d’optimiser les temps de calcul en augmentant simplement la puissance calculatoire. Malgré l’accélération non négligeable qu’ils fournissent, la mise en place de tels environnements matériels dédiés peut représenter un investissement coûteux.

1.2 Milieux participants

Un milieu est dit participant s’il transmet la lumière en affectant sa propagation, à l’inverse du vide où elle se propage en ligne droite. On regroupe dans ces milieux :

- l’atmosphère,
- les nuages, constitués de particules d’eau et/ou de glace,
- les fumées, constituées de particules de matière (e.g carbone, poussières),
- les gaz,
- le feu,
- les matières translucides (e.g. le marbre)...

Dans tous ces milieux, on retrouve un ou plusieurs des trois phénomènes qui affectent la propagation de la lumière : l'émission, l'absorption et la diffusion. Nous décrivons dans le chapitre 2 la théorie physique qui leur est associée.

De nos jours, un nombre croissant d'applications doivent prendre en compte les phénomènes liés à ces milieux. Leur simulation est utilisée dans des domaines variés, depuis le domaine militaire (e.g. simulation de télédétection, plumes de fumée sur les champs de batailles), jusqu'au domaine industriel (e.g. conception de phares efficaces pour les temps de brouillards). L'avantage de la simulation est qu'elle permet de recréer des conditions particulières d'observation qui seraient difficiles à mettre en place en réalité. Par exemple, dans l'aéronautique, elle permet la mise au point de systèmes infrarouges d'assistance au pilotage pour des conditions atmosphériques dégradées (brouillard, pluie...) sans pour autant mettre en péril un matériel coûteux et surtout la vie de pilotes d'essais.

Du point de vue informatique, la plupart des méthodes de synthèse d'images existantes négligent les phénomènes propres aux milieux participants. Dans un souci d'optimisation, ces méthodes supposent que la lumière se propage dans le vide entre les surfaces de la scène, on considère alors que la propagation s'effectue en ligne droite. Cette hypothèse permet notamment de négliger la diffusion, ce phénomène étant celui qui présente la plus grande complexité. Elle permet ainsi d'obtenir des performances satisfaisantes pour des applications visuelles (e.g. jeux vidéos, films d'animation, rendu architectural...). Néanmoins, la perte de réalisme et de précision qui en découle est déplorable. Par exemple, pour la visualisation d'une scène d'extérieur, l'absence de nuages dans le ciel nuit grandement au réalisme et rend plus difficile l'immersion de l'utilisateur. De même, négliger les gaz de combustion lors de l'étude d'un moteur peut engendrer des résultats de simulation complètement différents de la réalité.

C'est pourquoi un certain nombre de travaux ont proposé des méthodes qui permettent d'inclure des milieux participants dans la simulation. Dans la section 3.3, nous passons en revue ces méthodes afin d'étudier leurs avantages et leur inconvénients. Bien que certaines de ces méthodes offrent des résultats satisfaisants du point de vue de la qualité de la simulation, elles souffrent de trois problèmes de notre point de vue.

Tout d'abord, comme leurs équivalents surfaciques, ces méthodes ont été développées dans le cadre de la simulation visible. Elles négligent donc souvent la complexité spectrale des milieux (i.e. rendu RVB). Or celle-ci est beaucoup plus importante que dans le cas des surfaces. La figure 1.3 présente le spectre d'absorption du dioxyde de carbone. On note la présence de fortes

raies d'absorption pour certaines longueurs d'ondes. L'atmosphère, par exemple, est composée de nombreux gaz différents (azote, oxygène, dioxyde de carbone...) ainsi que de vapeur d'eau et de particules liées notamment à la pollution. Chacun de ses composants possède un spectre d'absorption similaire à celui de la figure 1.3. L'atmosphère possède donc un spectre résultant très irrégulier pour l'absorption. On retrouve également cette forte irrégularité dans le spectre de diffusion. La simulation RVB est incapable de traiter une telle irrégularité et peut donc rater un grand nombre de phénomènes, toutes les propriétés du milieu étant intégrées en bande grises pour les trois couleurs.

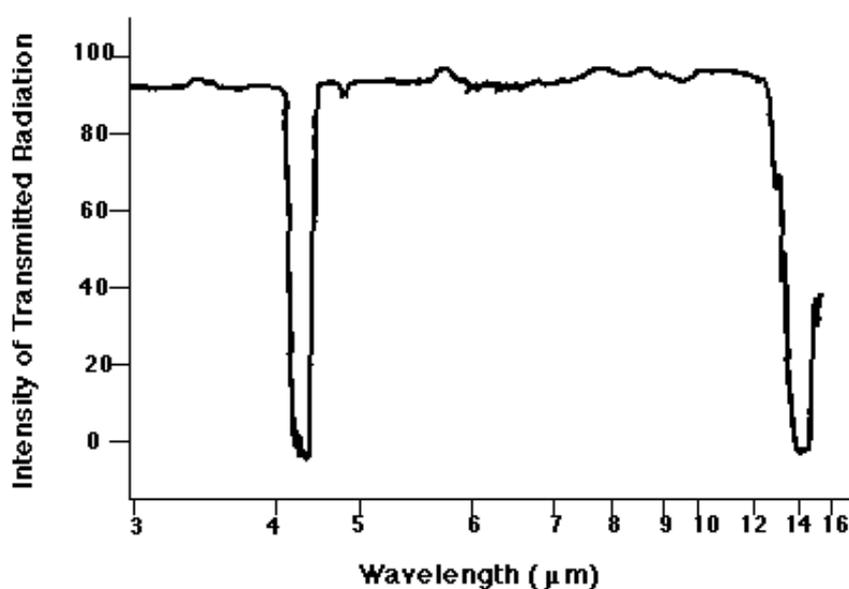


Figure 1.3. Spectre d'absorption du CO_2 en infrarouge.

Deuxièmement, ces méthodes font souvent des hypothèses drastiques sur la complexité du milieu simulé. Celles-ci portent souvent sur l'homogénéité et l'isotropie du milieu. L'homogénéité signifie que les propriétés du milieu ne varient pas en fonction de la position, l'isotropie du milieu signifie que l'indicatrice de diffusion (i.e. la fonction de phase) ne privilégie aucune direction de propagation de la lumière. Bien entendu, l'application de telles hypothèses affecte fortement le réalisme des images générées et encore plus la précision.

Finalement, le temps de calcul nécessaire à ces méthodes reste prohibitif pour des scènes complexes. Nous verrons que le phénomène le plus coûteux à simuler est la diffusion multiple, les autres phénomènes étant relativement simples à calculer.

Nos travaux se sont orientés vers la mise au point d'une méthode qui permette de lever le maximum d'hypothèses faites sur les propriétés du milieu, tout en offrant des temps de calcul

acceptables. Nous verrons, dans l'état de l'art présenté au chapitre 3, que la méthode des cartes de photons permet de lever les hypothèses sur la complexité du milieu. Néanmoins, nous montrerons que les performances obtenues par cette méthode ne sont pas satisfaisantes. Nous proposerons au chapitre 5 une méthode basée sur l'estimation de densité qui permet de diviser les temps de calcul par dix au minimum dans les pires cas d'utilisation.

Chapitre 2

Théorie du transfert radiatif

2.1 Introduction

Dans ce chapitre, nous présentons rapidement la théorie physique du transfert radiatif sur laquelle se basent les algorithmes de rendu de synthèse d'image pour résoudre la problématique présentée au chapitre 1. Ces modèles sont issus du monde de la physique. Dans ce cadre, ils ont été validés sur des cas d'école, c'est-à-dire sur des cas simples et unitaires dont on peut calculer la solution théorique (e.g. scène composée d'une source ponctuelle et d'une plaque, ou milieu participant infini à lames parallèles). La synthèse d'images permet de généraliser ces modèles en optimisant les calculs nécessaires, autorisant ainsi des simulations à grande échelle (i.e. scènes complexes).

Nous présentons tout d'abord l'équation du rendu qui unifie toutes les méthodes de synthèse d'image pour le rendu de scènes uniquement composées de surfaces. Nous présentons ensuite l'équation du transfert radiatif dans les milieux participants ainsi que leurs propriétés.

2.2 Equation du rendu

L'équation du rendu (EQR), mise au point par Kajiyama en 1986 [Kaj86], formalise le problème de l'illumination globale à partir d'une expression locale issue de la définition de la fonction de distribution de réflectance bidirectionnelle (cf. annexe A). Celle-ci a permis d'unifier tous les algorithmes de résolution existants. Pour une longueur d'onde λ , la luminance réfléchie L_r au

point x dans la direction ω_r s'exprime :

$$L_r(x, \omega_r, \lambda) = L_e(x, \omega_r, \lambda) + \int_{\Omega_i=2\pi} f_r(x, \omega_i, \omega_r, \lambda) L_i(\omega_i, \lambda) \cos\theta_i d\omega_i \quad (2.1)$$

Où L_e est la luminance émise au point x dans la direction ω_r , f_r est la fonction de réflectance bidirectionnelle (FDRB), L_i est la luminance arrivant au point x depuis la direction ω_i et Ω_i est l'espace des directions incidentes.

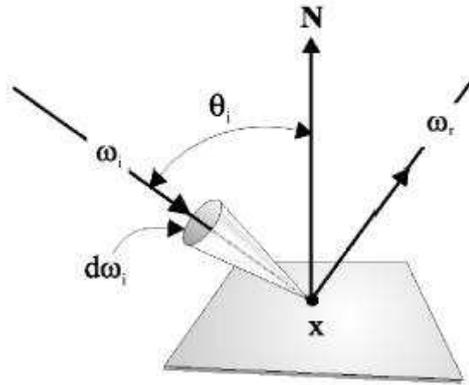


Figure 2.1. Géométrie associée à la formulation directionnelle de l'EQR.

Cette formulation "directionnelle" de l'équation du rendu est particulièrement adaptée aux méthodes orientées image telles que le lancer de rayon Monte Carlo. Néanmoins, les méthodes orientées scène telles que la radiosité préfèrent utiliser une expression plus "surfactive" de cette équation. En effet, la luminance incidente en un point x à une surface de la scène est la luminance émise ou réfléchie par les autres surfaces de la scène visible depuis x . On peut donc évaluer l'énergie totale provenant de tous les points x' appartenant aux surfaces de la scène. L'EQR s'écrit donc :

$$L_r(x, \omega_r, \lambda) = L_e(x, \omega_r, \lambda) + \int_{x'} f_r(x, \omega_i, \omega_r, \lambda) L_i(x', \omega_i, \lambda) \cos\theta_i \frac{\cos\theta_r}{\|x - x'\|^2} dA' \quad (2.2)$$

Généralement, on fait apparaître un terme purement géométrique évalué entre les deux points x et x' :

$$g(x, x') = d\omega_i \cos\theta_i = \frac{\cos\theta_r \cos\theta_i}{\|x - x'\|^2} \quad (2.3)$$

On introduit aussi la fonction de visibilité v entre les points x et x' , valant un s'ils sont directement visibles, zéro sinon. Finalement on obtient :

$$L_r(x, \omega_r, \lambda) = L_e(x, \omega_r, \lambda) + \int_{x'} f_r(x, \omega_i, \omega_r, \lambda) L_i(x', \omega_i, \lambda) g(x, x') v(x, x') dA' \quad (2.4)$$

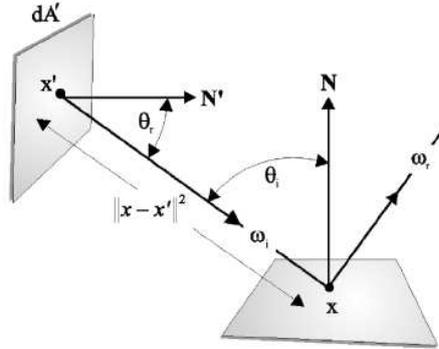


Figure 2.2. Géométrie associée à la formulation surfacique de l'EQR.

2.3 Milieux participants

2.3.1 Phénomènes élémentaires

Lorsqu'une onde lumineuse (énergétique de manière générale) se propage à l'intérieur d'un milieu participant, plusieurs phénomènes viennent modifier sa direction et l'énergie qu'elle transporte. Ces phénomènes optiques ont lieu de manière élémentaire, lorsque le rayon lumineux entre en collision avec un des constituants de base du milieu (molécules, aérosols,...). En théorie, on considère le milieu comme relativement dense et donc, il est plus naturel d'étudier les modifications optiques sur un chemin élémentaire l du rayon L dans le milieu (cf. figure 2.3). L'énergie du rayon sur ce chemin peut être réduite par les phénomènes d'absorption et de diffusion sortante et amplifiée par l'émission propre du milieu et la diffusion entrante. Nous allons étudier plus en détail chacun de ces phénomènes.

Remarque : Chacun des phénomènes ci-dessous est dépendant de la longueur d'onde. Dans un souci de clarté, nous considérerons implicitement les longueurs d'onde dans chacune des équations.

2.3.2 Absorption

Ce phénomène se caractérise par l'absorption d'une partie du rayonnement reçue par l'élément du milieu. Cela correspond au passage d'une partie de l'énergie ondulatoire (photons) à de

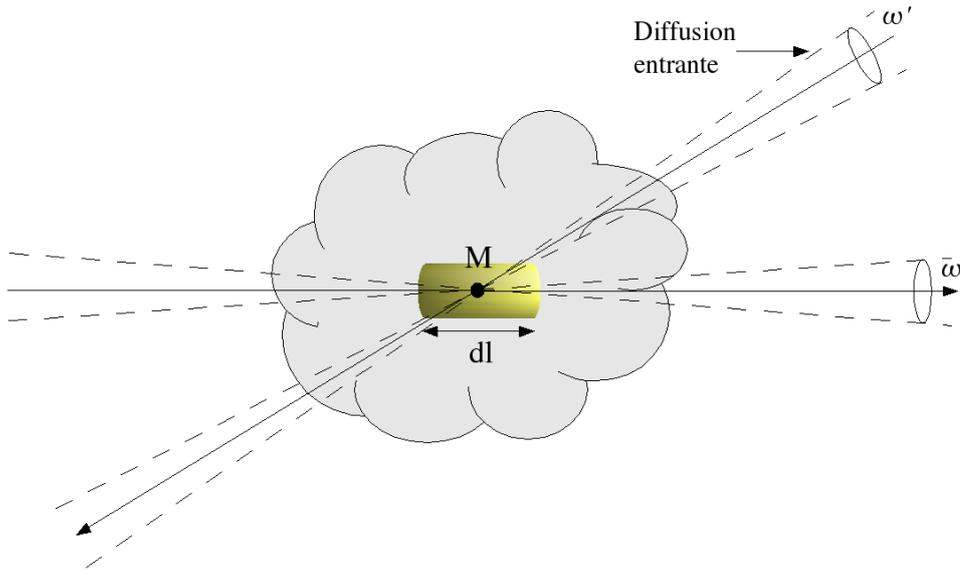


Figure 2.3. Géométrie liée à l'équation volumique du transfert radiatif dans son expression différentielle. On étudie la variation de luminance sur un chemin élémentaire dl du rayon L .

l'énergie vibratoire (phonons) propre à l'élément ayant reçu le rayonnement. En théorie, l'absorption est caractérisée par un coefficient σ_a (exprimé en m^{-1}). Il dépend de la température, de la pression, des concentrations des différents constituants du milieu et de la longueur d'onde du rayonnement incident (λ). La définition de ce coefficient nous permet d'établir la relation suivante :

$$dL(x, \omega) = -\sigma_a(x)L(x, \omega)dl \quad (2.5)$$

2.3.3 Emission

Le phénomène d'émission correspond, à l'inverse de l'absorption, au passage de l'énergie contenue dans l'élément du milieu sous forme d'énergie vibratoire, à une énergie de rayonnement. Ce phénomène est non nul pour les gaz chauds ou pour les particules de carbone (dans les flammes lumineuses). Comme l'absorption, il est indépendant de la direction du rayonnement considéré. A l'équilibre thermodynamique local, il peut être modélisé par l'équation suivante :

$$dL(x, \omega) = \sigma_a(x)L_e(x, \omega)dl \quad (2.6)$$

où $L_e(x, \omega)$ est la luminance du corps noir.

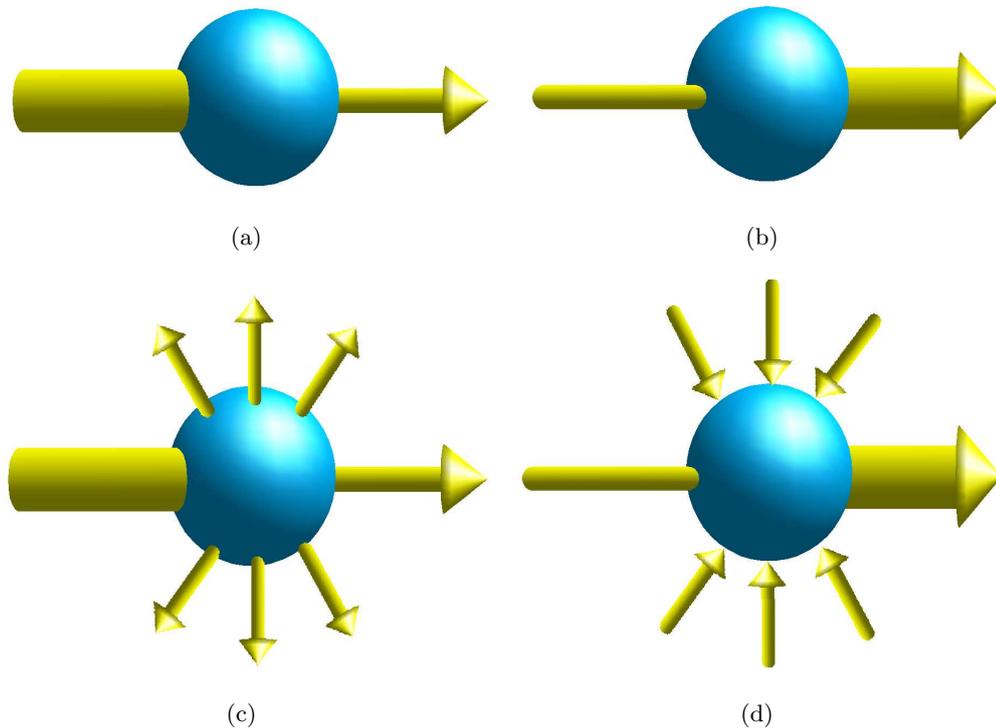


Figure 2.4. *Phénomènes modifiant la propagation de la lumière sur un trajet élémentaire dans un milieu participant. (a) Absorption. (b) Emission. (c) Diffusion sortante. (d) Diffusion entrante.*

2.3.4 Diffusion

Lorsqu'un rayonnement atteint un élément du milieu, il peut être diffusé. Cela signifie que l'énergie est déviée dans toutes les directions. La répartition angulaire de l'énergie lors de la diffusion est formalisé par des fonctions sphériques, les fonctions de phase, et un coefficient de diffusion σ_s . De manière générale, on s'intéresse à l'énergie reçue dans une direction donnée. Par commodité, on distingue alors deux cas théoriques : la diffusion sortante et la diffusion entrante.

2.3.4.1 Fonctions de phase

La définition de la fonction de phase est donnée dans l'annexe A.5. Nous décrivons ici les principales fonctions de phase qui nous intéressent.

Fonction de phase isotrope Il s'agit de la fonction de phase la plus simple. Comme son nom l'indique, elle caractérise la diffusion isotrope parfaite. L'énergie est diffusée uniformément dans toutes les directions de l'espace quelque soit la direction d'origine du rayonnement.

$$p(\theta) = \frac{1}{4\pi} \quad (2.7)$$

Fonction de phase de Henyey-Greenstein C'est la fonction de phase la plus souvent utilisée. C'est une fonction empirique introduite pour expliquer la diffusion due aux poussières intergalactiques, mais elle a ensuite été utilisée pour décrire la diffusion dans de nombreux types de milieux participants (océans, nuages, fumées, peau, pierre...). L'expression de la fonction est :

$$p(\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g \cos \theta)^{\frac{3}{2}}} \quad (2.8)$$

où g est le paramètre d'asymétrie compris entre -1 et 1. Une valeur positive de g décrit une diffusion en avant, tandis qu'une valeur négative décrit une diffusion vers l'arrière. Si $g = 0$, on retrouve la fonction de phase isotrope.

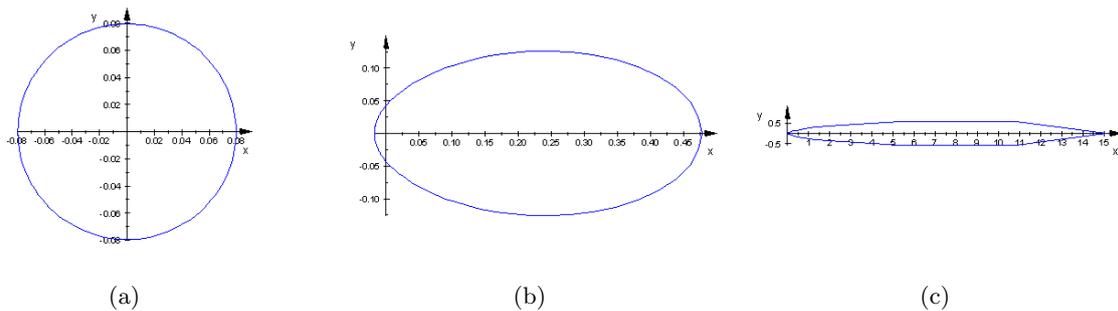


Figure 2.5. Fonction de phase de Henyey-Greenstein avec : (a) $g=0.0$, (b) $g=0.5$, (c) $g=0.9$.

Cette fonction est symétrique pour les valeurs négatives du paramètre d'asymétrie g .

Autres fonctions de phase Il existe beaucoup d'autres modèles de fonction de phase. Les plus connus sont le modèle de Rayleigh [BH83] et le modèle de Mie [Mie08].

Le modèle de Rayleigh est donné pour un milieu où les particules peuvent être assimilées à des sphères de taille très petite (de rayon inférieur à la longueur d'onde de la lumière). Dans ce modèle, la fonction de phase est presque uniforme, mais le coefficient de diffusion possède une forte dépendance spectrale où la lumière bleue est beaucoup plus diffusée que les autres. Elle est utilisée notamment pour les molécules de l'atmosphère, le ciel bleu et les couchers de soleil rouge.

Le modèle de Mie simule des milieux plus complexes. Il s'agit de milieux composés de sphères homogènes possédant des tailles arbitraires. Les formules de la théorie de Mie sont assez com-

plexes et difficiles à utiliser. Il est donc judicieux de ne les utiliser que lorsque l'on a vraiment besoin d'une grande précision pour la diffusion dans ce type de milieu. Une observation intéressante est que la diffusion de Mie est souvent caractérisée par une forte diffusion en avant et une petite diffusion en arrière. En image de synthèse, elle peut souvent être approchée par la somme de fonctions de Henyey-Greenstein.

2.3.4.2 Diffusion sortante

La diffusion sortante formalise la perte d'énergie due à la diffusion d'une partie du rayonnement dans toutes les directions de l'espace. Celle-ci est caractérisée par le coefficient de diffusion σ_s . De la même manière que pour le coefficient d'absorption, ce coefficient s'exprime en m^{-1} et dépend de la température, de la pression du milieu et de la longueur d'onde du rayonnement incident. Grâce à ce coefficient, on peut exprimer la variation de la luminance portée par le rayon et due à la diffusion sortante de la manière suivante :

$$dL(x, \omega) = -\sigma_s(x)L(x, \omega)dl \quad (2.9)$$

Les phénomènes d'absorption et de diffusion sortante ont les mêmes effets sur la luminance entrante. Ils se formalisent d'ailleurs de la même manière. Il semble donc logique de les regrouper par souci de clarté. On introduit donc un nouveau paramètre : le coefficient d'extinction $\sigma_t = \sigma_a + \sigma_s$.

2.3.4.3 Diffusion entrante

On considère un point x du milieu et la direction de rayonnement ω . La diffusion entrante est l'énergie provenant de toutes les directions (excepté la direction ω) qui est diffusée par le point x dans la direction ω . Il peut s'agir, soit de l'énergie provenant directement des sources lumineuses (diffusion simple), soit de l'énergie qui a été diffusée par les autres points du milieu (diffusion multiple). Ce phénomène correspond donc à une augmentation de l'énergie le long de la direction de propagation ω .

Il faut intégrer toutes les luminances incidentes à l'élément et leur diffusion dans la direction considérée pour obtenir finalement le terme modélisant la diffusion entrante :

$$dL(x, \omega) = \frac{\sigma_s(x)dl}{4\pi} \int_{\Omega=4\pi} p(x, \omega, \omega')L(x, \omega')d\omega' \quad (2.10)$$

Le coefficient σ_s est identique à celui utilisé pour la diffusion sortante et représente la capacité de diffusion de la particule en fonction de la température, de la pression du milieu, de la longueur d'onde du rayonnement incident et des concentrations des différents constituants mis en jeu.

2.4 Equation volumique du transfert radiatif

L'équation volumique du transfert radiatif (ou ETR) [Cha50] correspond à l'intégration le long d'un chemin l de direction ω de l'ensemble des phénomènes vus précédemment (figure 2.3). Elle exprime la variation de luminance due à la traversée du milieu participant sur ce chemin. On peut la trouver sous son expression différentielle :

$$\begin{aligned} \frac{dL(x, \omega)}{dl} &= -\sigma_a(x)L(x, \omega) + \sigma_a(x)L_e(x, \omega) - \sigma_s(x)L(x, \omega) \\ &\quad + \frac{\sigma_s(x)}{4\pi} \int_{\Omega=4\pi} p(x, \omega, \omega')L(x, \omega')d\omega' \end{aligned} \quad (2.11)$$

On peut intégrer cette équation sur un chemin lumineux $l = [x_n, x_0]$ de direction ω , où x_n est le point d'entrée dans le milieu participant et x_0 le point de sortie. Si l'on suppose que lorsqu'on n'est pas dans le milieu, on est dans le vide, la luminance au point d'observation x dans la direction ω est égale à la luminance au point d'entrée x_n . On a alors l'expression intégrée de l'équation volumique du transfert radiatif :

$$\begin{aligned} L(x, \omega) &= e^{-\tau(x_0, x_n)}L(x_0, \omega) + \int_{x_0}^{x_n} e^{-\tau(u, x_n)}\sigma_a(u)L_e(u, \omega)du \\ &\quad + \int_{x_0}^{x_n} e^{-\tau(u, x_n)}\frac{\sigma_s(u)}{4\pi} \int_{\Omega=4\pi} p(u, \omega, \omega')L(u, \omega')d\omega' du \end{aligned} \quad (2.12)$$

où $\tau(a, b) = \int_a^b \sigma_t(x)dx$ est l'épaisseur optique du chemin $[a, b]$ et $\sigma_t = \sigma_a + \sigma_s$ est le coefficient d'extinction.

2.5 Conclusion

Nous avons présenté ici l'équation du rendu (2.1) et l'équation volumique du transfert radiatif (2.12) qui sont les équations qui uniformisent les méthodes de synthèse d'image. En effet, toutes ces méthodes tentent de résoudre ces équations le plus rapidement possible tout en gardant une précision acceptable dans les solutions obtenues. Dans le chapitre 3, nous allons présenter un inventaire des méthodes d'illumination globale tant pour le rendu de surfaces que pour celui de

milieux participants, ce qui correspond aux deux points de notre problématique. Nous comparons ces différentes méthodes et justifions ainsi le choix de l'adaptation de la méthode des cartes de photons au rendu infrarouge.

Chapitre 3

Calcul de l'illumination globale

3.1 Introduction

A l'origine du problème de l'illumination globale en synthèse d'image, on trouve la synthèse d'image physiquement réaliste. Il s'agit d'une branche de l'infographie. Son but était de générer des images qu'un observateur humain serait incapable de différencier d'une photographie. Cependant la synthèse d'image physiquement réaliste est un problème complexe car elle nécessite une modélisation précise de la géométrie de la scène, des modèles de matériaux réalistes et un mécanisme d'éclairage physiquement correct, l'observateur étant très sensible aux artefacts dus aux simplifications nécessaires pour l'optimisation.

Aujourd'hui, la puissance des ordinateurs permet de traiter des scènes de plus en plus complexes tant au niveau de la complexité géométrique des scènes, qu'au niveau des modèles de matériaux et des modèles d'éclairage. La description géométrique des scènes peut être très précise si l'on se donne les moyens. Le problème est alors uniquement un problème de temps et de talent de l'infographiste. Néanmoins, de nombreuses recherches ont été menées pour faciliter et accélérer le travail d'infographie. Deux thèmes de recherche dans ce domaine sont, par exemple, la modélisation par surfaces implicites [BW86], et la modélisation déclarative [LDMP89]. La modélisation des matériaux suscite également beaucoup d'intérêt. Un grand nombre de modèles ont été mis au point. La plupart du temps, en synthèse d'image non réaliste, des modèles simples sont utilisés (Phong, Gouraud). Par contre des modèles beaucoup plus réalistes ont été développés, notamment des modèles qui permettent de prendre en compte des données réelles mesurées. Mais, pour obtenir une représentation fidèle des matériaux mesurés, on a besoin de gros volumes de données dont la manipulation pose problème. Il est donc nécessaire de compres-

ser les données. On trouve par exemple un modèle de compression par ondelettes de la FDRB [Cla03]. La majorité des modèles de matériaux ne considèrent que la FDRB pour la réflexion. Cependant, pour beaucoup de matériaux (par exemple les textiles), la réflexion varie avec la position sur la surface. Le modèle des fonctions de textures bidirectionnelles (*Bidirectional Texture Function i.e. BTF*, [MMS⁺04]) permet de traiter ce type de matériau de manière plus réaliste.

Bien que tous ces travaux soient très prometteurs, ils ne permettront d'atteindre le niveau de réalisme recherché que s'ils sont complétés par des modèles d'éclairages complexes. C'est dans ce cadre qu'apparaissent les modèles d'illumination globale. Leur but est de simuler rapidement et de manière aussi fidèle que possible tous les transferts radiatifs en fonction de la géométrie et des matériaux de la scène. On parle d'illumination globale car on tente de prendre en compte toutes les inter-réflexions entre tous les éléments de la scène. En effet, si l'on considère un point de calcul de l'illumination, on s'aperçoit que l'énergie reçue par ce point provient d'une part des sources lumineuses mais aussi, indirectement, de tous les autres points de la scène. D'où la complexité du problème à résoudre. Dans ce cas, la modification de l'éclairage en un point de la scène influe sur l'ensemble de la scène, contrairement aux modèles d'illumination locale qui considèrent uniquement l'énergie provenant de sources.

Ce chapitre établit un état de l'art des méthodes d'illuminations globales. Ces méthodes tentent de résoudre les équations présentées au chapitre 2. Nous exposons tout d'abord les méthodes qui traitent les scènes où l'on suppose que les surfaces sont plongées dans le vide. Ensuite, nous décrivons les méthodes qui traitent les milieux participants. Dans le premier cas, nous faisons la distinction entre les méthodes à base des radiosité et les méthodes à base de lancer de rayons. Dans le deuxième cas, nous les classifions entre méthodes déterministes et stochastiques.

3.2 Surfaces

3.2.1 Méthodes de Radiosité

Equation de la Radiosité La méthode de radiosité est historiquement la première méthode de résolution de l'éclairage global d'une scène. Elle a été introduite par Goral en 1984 [GTGB84] et se base sur la théorie du transfert radiatif de chaleur adaptée à la lumière. Le problème de l'illumination globale étant très complexe, des hypothèses fortes ont été prises afin de simplifier les calculs :

- Les sources lumineuses sont supposées isotropes et homogènes.
- Les surfaces sont lambertiennes (parfaitement diffuses).

Ainsi la composante directionnelle de l'équation du rendu est supprimée :

$$L_r(x) = L_e(x) + \int_{x'} f_r(x) L_i(x') g(x, x') v(x, x') dA' \quad (3.1)$$

De plus on remarque que, pour une surface diffuse, la luminance est égale à la radiosité (notée B) divisée par un facteur π (dû à l'intégration sur l'espace des directions). Par convention, la puissance émise par unité d'aire est appelée exitance (notée E). De même, la FDRB est égale à la réflectance divisée par π , et est donc constante. En la mettant en facteur, on a pour une longueur d'onde donnée :

$$B(x) = E(x) + \frac{\rho(x)}{\pi} \int_{x'} B(x') g(x, x') v(x, x') dA' \quad (3.2)$$

En regroupant g, v et le facteur π , on obtient le "noyau géométrique" :

$$K(x, x') = \frac{g(x, x') v(x, x')}{\pi} \quad (3.3)$$

Puis l'équation de la radiosité :

$$B(x) = E(x) + \rho(x) \int_{x'} B(x') K(x, x') dA' \quad (3.4)$$

Résolution par la méthode des éléments finis Généralement, la méthode des éléments finis et plus particulièrement de Garlekin (base de fonctions orthogonales) sert à la résolution de cette équation. La méthode des éléments finis [Zie89] représente une fonction complexe f en subdivisant son domaine de définition en n éléments e_i , sur lesquels elle est approchée par une somme linéaire (projection sur une base de fonctions $\{\gamma_i\}$) :

$$f(x) \approx \bar{f}(x) = \sum_i \sum_j f_j^i \gamma_j^i(x) \quad (3.5)$$

Cette technique a été appliquée pour projeter la fonction de radiosité [Hec93] [BW96]. En substituant l'approximation à B dans l'équation de la radiosité 3.4, on obtient un système d'équations linéaires :

$$B_i^j = E_i^j + \sum_{i', j'} K_{j, j'}^{i, i'} B_{j'}^{i'} \quad (3.6)$$

Les E_i^j étant les coefficients de projection de $E(x)$:

$$E_i^j = \int_x \gamma_j^i(x) E(x) dA \quad (3.7)$$

et $K_{j,j'}^{i,i'}$ le facteur de forme généralisé :

$$K_{j,j'}^{i,i'} = \int_x \gamma_j^i(x) \int_{x'} \rho(x) K(x, x') \gamma_{j'}^{i'}(x') dA dA' \quad (3.8)$$

Les étapes de formulation et de résolution d'un problème de radiosité générique consistent donc à :

- subdiviser les surfaces de la scène en éléments A_i ;
- choisir une base de fonctions γ pour la projection ;
- calculer le facteur de forme généralisé K entre chaque élément de la scène ;
- résoudre le système d'équations linéaires (les inconnues étant les coefficients de projection) ;
- reconstruire la fonction de radiosité grâce à ces coefficients ;
- calculer une image à partir de la fonction de radiosité obtenue.

Le problème principal de la radiosité par éléments finis est sa complexité en terme de temps de calcul et de coût mémoire. Pour n_e éléments et des fonctions d'ordre n_0 , la résolution du système par inversion matricielle directe a une complexité en $O(n^3)$ avec $n = n_e n_0$. En fait, cette matrice est creuse et des méthodes plus performantes peuvent être utilisées [GCS93] : méthode de Gauss-Seidel ou relaxation de Southwell par exemple. Leur complexité calculatoire est $O(n^2)$, et $O(n^2)$ (respectivement $O(n)$) en mémoire. A cause des nombreuses discontinuités de la radiosité (dues notamment aux ombres), des fonctions d'ordre supérieur sont rarement utilisées. C'est pourquoi les fonctions constantes sur les éléments sont privilégiées [GTGB84], ce qui simplifie encore plus l'équation de radiosité :

$$B_i = E_i + \rho_i \sum_j F_j^i B_j \quad (3.9)$$

avec le facteur de forme simple :

$$F_j^i = \frac{1}{A} \int_x \int_{x'} K(x, x') dA dA' = \frac{1}{A_i} \int_{A_i} \int_{A_j} K(x, x') dA_i dA_j \quad (3.10)$$

Radiosité hiérarchique De nombreuses méthodes dites hiérarchiques ont été développées afin de réduire la complexité de l'algorithme de radiosité. Le principe de ces méthodes est que le transfert énergétique entre deux surfaces peut se faire à différents niveaux de résolution. Ainsi,

entre deux surfaces proches, il est nécessaire de discrétiser finement les surfaces pour obtenir une représentation suffisamment précise du transfert. Par contre, si les surfaces sont éloignées, une discrétisation plus grossière est suffisante. La première approche dans ce sens a été mise en oeuvre par Hanrahan en 1991 [HSA91] sous le nom de radiosité hiérarchique. L'intérêt est de diminuer la complexité en $O(k^2 + n)$ où k est le nombre de surfaces initiales et n le nombre total d'éléments discrets. Il a été montré par Gortler que ce principe est équivalent à une résolution de la radiosité en utilisant une base de fonctions par ondelettes [GSCH93].

Cependant, la complexité de la radiosité hiérarchique reste quadratique par rapport au nombre de surfaces initiales de la scène. Ceci pose problème lorsque l'algorithme est appliqué à des scènes complexes. Pour résoudre ce problème, des méthodes de regroupement ou "clustering" ont été proposées par Smits [Smi94] et par Sillion [Sil94]. Ces méthodes consistent à classer les surfaces initiales en groupes ou "clusters". Les échanges énergétiques peuvent alors se faire entre groupes de surfaces ou entre éléments et groupes de surfaces. Cela reste néanmoins très difficile à mettre en oeuvre pour des scènes quelconques [HDSD99].

Une des difficultés principales de la radiosité hiérarchique (avec ou sans regroupement) est de décider à quel niveau doit s'effectuer le transfert d'énergie. De nombreux "oracles" ont été mis au point pour prendre cette décision. L'idée générale est de créer un transfert à un niveau si l'erreur estimée est inférieure à un certain seuil. L'erreur est généralement estimée localement en prenant en compte la variation du facteur de forme sur le récepteur et la variation de la radiosité sur la source [GH96]. Une estimation de l'erreur globale a aussi été proposée [LSG94] mais elle s'est avérée trop pessimiste pour être utilisable. En général, le problème est donc que l'erreur en radiosité hiérarchique est difficilement contrôlable puisqu'elle est liée à des paramètres dépendant fortement de la scène. Par exemple, les temps de calcul ne varient pas linéairement avec le seuil d'erreur utilisé.

Radiosité Monte Carlo La méthode de Monte Carlo est une méthode d'intégration stochastique. Elle peut être utilisée de quatre différentes façons pour résoudre les systèmes d'équations linéaires de la radiosité :

- Les coefficients du système à résoudre peuvent être précalculés par la méthode de Monte Carlo. Une méthode traditionnelle (directe ou itérative) est ensuite utilisée pour résoudre le système approché obtenu.
- Dans les méthodes de relaxation stochastique, les produits matrice-vecteur sont estimés

simultanément au lieu d'être traités un par un.

- Les méthodes de tracer de chemins aléatoires estiment la totalité de l'expansion des séries de Neumann de la solution (si elle converge) par la méthode de Monte Carlo.
- De nombreuses autres méthodes de Monte Carlo estiment la solution en résolvant un problème équivalent par la méthode de Monte Carlo.

Plus d'informations sur ces méthodes peuvent être trouvées dans la thèse de Bekaert [BCW99].

Extensions de la méthode de radiosit  Les hypoth ses fortes sur les sources lumineuses et les mat riaux permettent de simplifier le probl me du transfert radiatif. Cependant, le r alisme de la simulation obtenue laisse un peu   d sireux. De nombreuses m thodes se sont donc attel es   augmenter le r alisme des m thodes de radiosit , notamment par la gestion de FDRB plus complexes [ICG86, SAWG91, GSCH93, CSSD96, SSSS98]. M me si le r alisme des r sultats obtenus augmente, des probl mes persistent. Par exemple, les surfaces courbes brillantes ne sont pas g r es de mani re satisfaisante.

Conclusion Les algorithmes de radiosit  sont performants pour des sc nes simples mais leur efficacit  chute lorsque la complexit  augmente. D'une part, la prise en compte de mat riaux complexes (et notamment sp culaires) demande le rajout de traitements sp cifiques. D'autre part, la discr tisation des surfaces de la sc ne est un probl me non encore r solu de fa on satisfaisante. Le maillage de discontinuit  [LTG92] est une solution envisageable mais tr s co teuse qui augmente de mani re cons quente la complexit  du mod le. Cette discr tisation (accompagn e du calcul de l'erreur commise) dans l'espace sc ne s'effectue ind pendamment de l'observateur. Le maillage obtenu n'est donc pas toujours adapt    son point de vue, par exemple lorsqu'il se rapproche tr s pr s des surfaces. Cette d pendance   la complexit  de la sc ne rend ces m thodes inapplicables dans notre cas, car nous souhaitons traiter des sc nes quelconques. N anmoins, le principe de pr calcul de l' clairage est int ressant car il permet de pr parer les donn es   l'avance et d'optimiser ensuite le temps de rendu. Pour plus de pr cision sur l'ensemble de ces m thodes, le lecteur peut se r f rer   [CW93, Sil94, Sta00].

Dans la section suivante, nous pr sentons les m thodes   base de lancer de rayons. Ces m thodes  tant bas es image (i.e. les calculs sont effectu s dans l'espace image), elle sont ind pendantes de la complexit  de la sc ne, ce qui est dans notre int r t.

3.2.2 Méthodes de Lancer de rayons

La méthode du lancer de rayons a été introduite par Whitted en 1980 [Whi80] en tant qu'extension de la méthode d'élimination des parties cachées d'Appel [App68]. Cette méthode est certainement la méthode la plus populaire de la synthèse d'images réaliste bien qu'elle ne traite pas tous les transferts lumineux possible et notamment l'éclairage indirect. De part sa simplicité de mise en oeuvre et de compréhension, cette méthode a été rapidement utilisée dans des applications industrielles et grand public contrairement aux méthodes de radiosité qui tardent encore à s'imposer. En effet, la méthode de radiosité est moins intuitive que le lancer de rayons, au sens où il est très difficile de connaître l'effet des paramètres sur la solution obtenue.

Principe du Lancer de rayons Basée sur l'optique géométrique, la méthode consiste à suivre la propagation inverse de la lumière. Un rayon (ou demi-droite) est tracé depuis la position de l'observateur à travers chaque pixel de l'image. Lorsque le rayon a une intersection avec une surface de la scène, l'éclairage reçu par le point d'intersection est calculé et est affecté au pixel. On distingue trois composantes dans le calcul de l'éclairage : l'éclairage direct, l'éclairage réfléchi et l'éclairage transmis. L'éclairage direct est calculé en traçant un rayon d'ombre vers chaque source lumineuse. La contribution de la source est ajoutée si elle est visible depuis le point d'intersection, c'est-à-dire si le rayon d'ombre ne rencontre pas d'obstacle. Les éclairages réfléchis et réfractés sont calculés en lançant deux nouveaux rayons et en calculant récursivement l'éclairage "récolté" par ces rayons. La direction des rayons réfléchis et réfractés est calculée en utilisant la loi de Descartes.

Cette méthode a été développée de manière intuitive avant l'introduction de l'EQR par Kajiyi [Kaj86]. Néanmoins, il est possible de montrer que le lancer de rayons respecte grossièrement cette équation.

Un seul rayon est tracé pour évaluer chaque composante de l'éclairage, ce qui revient à n'effectuer qu'une estimation ponctuelle de l'éclairage. Ceci entraîne des problèmes de crénelage puisque l'éclairage est une fonction continue et non pas discrète. L'algorithme est limité aux sources ponctuelles et aux réflexions spéculaires parfaites (miroir). Une extension a été proposée par Cook en 1984 [CPC84] appelée lancer de rayons distribué. Cette méthode consiste à lancer plusieurs rayons au lieu d'un seul pour évaluer chaque composante. Elle permet de résoudre les problèmes de crénelage de l'image, de gérer les sources étendues, de prendre en compte des surfaces non purement spéculaires, etc...

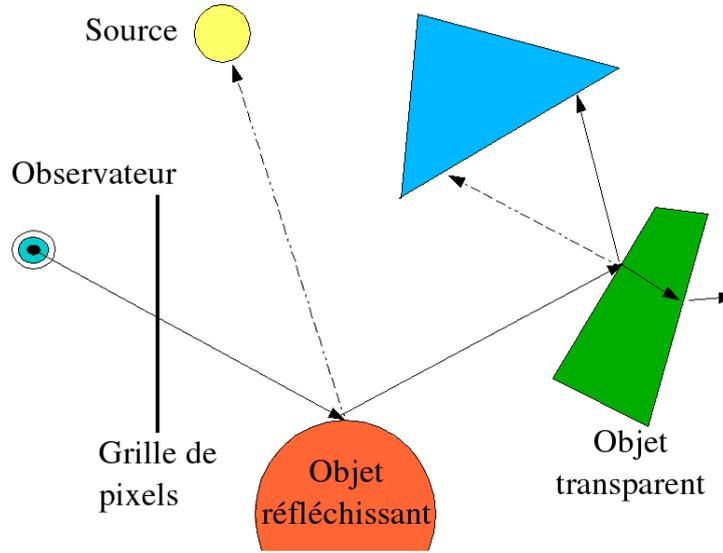


Figure 3.1. Principe du lancer de rayons

Lancer de rayons stochastique Dans cette méthode [AK94], la méthode de Monte Carlo est utilisée pour résoudre l'EQR. Elle consiste à choisir N échantillons (directions) selon une fonction de densité de probabilité (FDP) p dans l'hémisphère supérieur et à approcher l'intégrale par un estimateur :

$$L_r(x, \omega_r, \lambda) = L_e(x, \omega_r, \lambda) + \frac{1}{N} \sum_{j=1}^N \frac{f_r(x, \omega_i^j, \omega_r, \lambda)}{p(\omega_i^j)} L_i(x, \omega_i^j, \lambda) \cos \theta_i^j \quad (3.11)$$

Cette procédure, basée sur un échantillonnage aléatoire guidé, ne converge vers la solution que lorsque le nombre d'échantillons tend vers l'infini. Or, comme utiliser une infinité d'échantillons est impossible, nous nous contentons de N échantillons, ce qui introduit de la variance, c'est-à-dire un écart avec la solution réelle. Si on note x la luminance d'un pixel de l'image, l'expression de la variance est donnée par :

$$Var(x) = E(x^2) - (E(x))^2$$

Elle mesure l'écart entre les valeurs prises par x et la valeur moyenne $E(x)$. Théoriquement, on montre que celle-ci est proportionnelle à $\frac{1}{\sqrt{N}}$. Par exemple, pour diviser l'erreur par deux, on doit utiliser quatre fois plus d'échantillons. Cette variance est visible sous forme de bruit sur l'image générée. La luminance est estimée récursivement : la même équation est utilisée pour calculer L_i . Le processus récursif est stoppé après un nombre fixe d'itérations ou quand la contribution de la luminance calculée est très faible par rapport à l'énergie totale dans le cas du lancer de rayons déterministe. Cependant, ce procédé introduit un biais dans la simulation, c'est-à-dire

que la solution ne converge pas vers la solution exacte. C'est pourquoi on utilise la méthode de la Roulette Russe [SG69, AK94] pour terminer la récursion. Cette méthode se base sur la réflectance ρ du matériau (FDRB intégrée) pour déterminer si le rayon est réfléchi ou absorbé. La réflectance donne le ratio d'énergie réfléchi par la surface, de manière équivalente, la probabilité qu'un rayon soit réfléchi (ρ) ou absorbé ($1 - \rho$). En effet, d'après le principe de conservation de l'énergie on a $0 < \rho < 1$. La roulette russe permet de favoriser les chemins apportant le plus d'informations tout en assurant une répartition équitable de ces chemins. L'avantage de cette méthode est d'assurer que tous les chemins lumineux sont finis et que la simulation converge vers la solution réelle. Toutefois, l'utilisation de la Roulette Russe introduit un biais supplémentaire dans l'estimateur.

Selon la théorie de Monte Carlo, le choix de la densité de probabilité p pour échantillonner l'espace des directions a un impact très important sur l'erreur commise lors de l'estimation de la luminance. Dans l'idéal, il faut que la densité de probabilité suive parfaitement la répartition de la fonction intégrée : $p \propto f_r L_t \cos \theta$. En effet, on comprend intuitivement que l'intégrale sera plus importante aux endroits où la fonction elle-même l'est. Il est donc intéressant d'obtenir une plus grande précision (donc plus d'échantillons) dans ces zones. Mais L_t étant inconnue (c'est la valeur que nous voulons évaluer), nous ne pouvons nous baser au mieux que sur $f_r \cos \theta$. On appelle "échantillonnage par importance" l'utilisation d'une telle FDP pour la génération des échantillons qui permet d'abaisser considérablement la variance. Rigoureusement, p peut être obtenue par inversion mathématique de la fonction $f_r \cos \theta$. C'est pourquoi elle a déjà été calculée et utilisée dans le cas de modèles de FDRB simples (Lambert, Phong, Ward, Lafortune). Par contre, elle devient impossible à calculer pour des modèles complexes ou numériques. La méthode de rejet est alors utilisée au prix d'une convergence déplorable, i.e. une variance élevée [Gla95].

Une optimisation supplémentaire consiste à découper le domaine d'intégration en deux : la luminance provenant des sources (éclairage direct) et la luminance provenant du reste de la scène (éclairage indirect). Il s'agit ensuite d'échantillonner directement les sources de lumière. A part pour les sources lumineuses, dont la luminance est L_e , on obtient pour une surface quelconque :

$$L_r(x, \omega_r, \lambda) = \frac{1}{N_d} \sum_{j=1}^{N_d} \frac{f_r(x, \omega_i^j, \omega_r, \lambda)}{p(\omega_i^j)} L_e(-\omega_i^j, \lambda) \cos \theta_i^j$$

$$+ \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{f_r(x, \omega_i^j, \omega_r, \lambda)}{p(\omega_i^j)} L_i(\omega_i^j, \lambda) \cos \theta_i^j \quad (3.12)$$

où N_d est le nombre de directions générées dans l'angle solide sustenté par les diverses sources de la scène et N_i est le nombre de directions générées dans le domaine angulaire restant. Générer une direction ω_i^j vers une source lumineuse est équivalent à choisir un point x'_j sur cette source (qui est en fait l'intersection entre la direction et la source). Néanmoins, comme générer une direction pointant vers une surface est généralement complexe, on préfère intégrer sur la surface des sources :

$$\begin{aligned} L_r(x, \omega_r, \lambda) &= \frac{1}{N_d} \sum_{j=1}^{N_d} \frac{f_r(x, \omega_i^j, \omega_r, \lambda)}{p(x'_j)} L_e(x'_j, -\omega_i^j, \lambda) \cos \theta_i \frac{\cos \theta_r}{\|x - x'_j\|^2} \\ &+ \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{f_r(x, \omega_i^j, \omega_r, \lambda)}{p(\omega_i^j)} L_i(-\omega_i^j, \lambda) \cos \theta_i^j \end{aligned} \quad (3.13)$$

Sur-échantillonnage des pixels Dans l'algorithme de lancer de rayons direct, le but est d'évaluer directement le flux énergétique Φ reçu par le capteur. Un capteur reçoit la lumière sur toute sa surface d'aire A_c et sur tout son domaine angulaire Ω_c . On a donc :

$$\Phi = \int_{A_c} \int_{\Omega_c} L(x, \omega) \cos \theta d\omega dx \quad (3.14)$$

où L est la luminance incidente au point x depuis la direction ω et θ l'angle entre ω et la normale à la surface du capteur en x . La surface du capteur, ainsi que son domaine angulaire, sont échantillonnés et le flux est estimé grâce à une intégration de Monte Carlo. Généralement, on emploie un modèle de caméra sténopé représentant l'œil comme ponctuel, donc comme un capteur non surfacique. Le plan image (discrétisé en pixels) fournit le domaine angulaire visible. A partir de N_p échantillons (positions dans le pixel), dont chacun définit une direction unique et donc un rayon "primaire", la luminance en entrée du capteur est évaluée grâce à l'équation précédente. Pour cela, à chaque surface rencontrée, il faut générer un ensemble de rayons vers les sources lumineuses (rayons d'ombre) puis un ensemble de rayons secondaires évalués récursivement. L'intégration sur les pixels peut aussi être optimisée grâce à un schéma d'échantillonnage particulier (stratifié par exemple).

Tracer de chemins Un cas particulier du lancer de rayons Monte Carlo est le tracer de chemins. Il a été proposé par Kajiya [Kaj86] en même temps que la formulation de l'EQR.

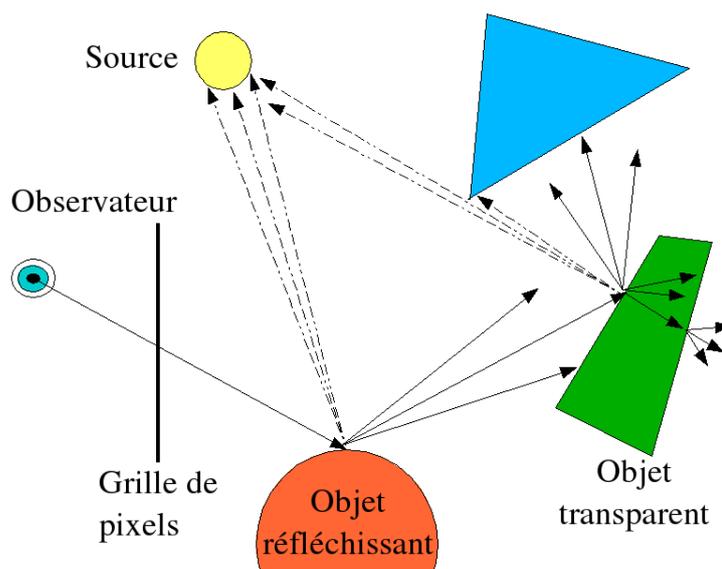


Figure 3.2. Principe du lancer de rayons stochastique.

Intuitivement, la solution est la même que le lancer de rayons mais avec un guidage stochastique des rayons. Il s'agit donc d'un lancer de rayons Monte Carlo où l'on utilise qu'un seul échantillon ($N_d = N_i = 1$). Pour chaque rayon primaire, on génère de façon stochastique (guidé ou non par une FDP) un seul rayon réfléchi (ou transmis) à chaque point d'intersection x_i . Le rayon primaire et l'ensemble des réflexions/transmissions forment donc un "chemin" lumineux $x_0x_1\dots x_n$ dans la scène.

Cependant, il est nécessaire de suivre un très grand nombre de chemins pour chaque pixel afin d'obtenir une solution précise et peu bruitée, la probabilité qu'un chemin lumineux passe par une source lumineuse étant réduite (i.e. le chemin ne rapporte aucune énergie). Ce nombre dépendant de la complexité de la scène, des propriétés des surfaces et des sources (N_p valant souvent plusieurs milliers). L'avantage par rapport au lancer de rayons distribué est qu'il n'y a pas une explosion exponentielle du nombre de rayons à traiter. Pour le tracer de chemins, la complexité est constante. De plus, pour des raisons de cohérence, suivre un chemin de nombreuses fois par pixel est plus performant que d'en suivre plusieurs en une même opération. Le tracé de chemin revient à résoudre l'EQR par "marche aléatoire".

Lancer de particules Le lancer de particules a été initié par Arvo en 1986 [Arv86] sous le titre de "lancer de rayons inverse". La méthode a été formalisée par Pattanaik en 1992 [PM92] et est l'approche duale du tracer de chemins, au sens où les chemins sont tracés en partant des sources plutôt que du capteur. L'avantage par rapport au tracer de chemins est que, comme les

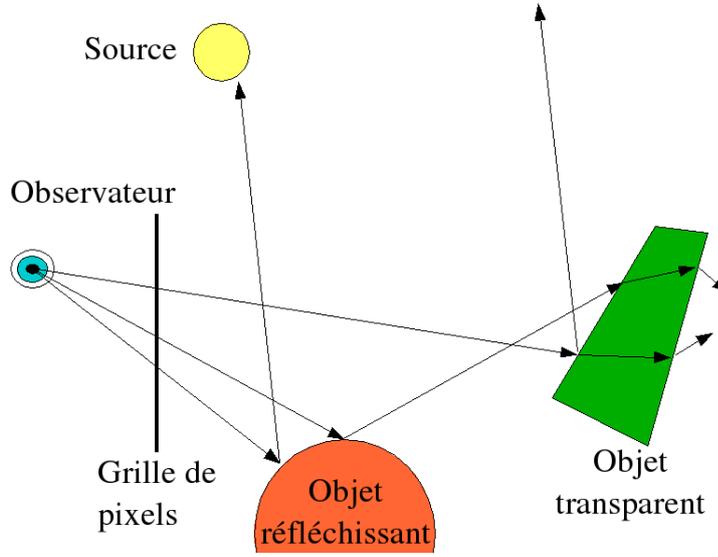


Figure 3.3. Principe du tracer de chemins.

rayons partent des sources, on est certain que chaque chemin calculé transporte de l'énergie. On introduit le concept de "particule" car cette méthode revient à suivre le trajet direct d'une particule de lumière au lieu du trajet inverse. Pour cela, Pattanaik introduit une équation similaire à l'EQR dans laquelle l'inconnue n'est plus la luminance, mais la fonction "potentiel" (W) décrivant la réponse du capteur. L'équation du potentiel (EQP) est la suivante :

$$W(x, \omega_r, \lambda) = W_e(x, \omega_r, \lambda) + \int_{\Omega_r} f_r(x, \omega_i, \omega_r, \lambda) W(\omega_i, \lambda) \cos \theta_i d\omega_i \quad (3.15)$$

Une présentation formelle et rigoureuse de cette équation peut être trouvée dans [Dut96]. Expliquons simplement ici que le domaine du capteur est un sous-ensemble de l'ensemble des surfaces et des directions de la scène : $A_c \times \Omega_c \subset A \times \Omega$. On peut donc écrire :

$$\Phi = \int_A \int_{\Omega} L(x, \omega) W_e(x, \omega) \cos \theta d\omega dx \quad (3.16)$$

où W_e est le "potentiel initial". $W_e = 1$ si $x \in A_c$ et $\omega \in \Omega_c$, $W_e = 0$ sinon. Pattanaik montre que la fonction de potentiel obéit aux mêmes règles que la luminance (invariance le long d'un rayon dans le vide, etc.) et en appliquant une formulation duale, il obtient :

$$\Phi = \int_A \int_{\Omega} L(x, -\omega) W(x, \omega) \cos \theta d\omega dx \quad (3.17)$$

avec W défini par l'équation précédente. Par inversion, on peut définir la fonction potentiel par :

$$W(x, \omega_r) = \frac{d^2 \Phi}{L(x, \omega_i) dA d\omega_i} \quad (3.18)$$

Cette équation permet de résoudre le problème de l'illumination globale non pas à partir du capteur mais depuis les sources. Il faut noter que c'est le principe de réciprocité, couramment admis en synthèse d'image, qui permet d'utiliser indifféremment la FDRB dans le sens observateur \rightarrow lumière ou lumière \rightarrow observateur. Pour l'estimateur primaire de l'équation 3.16, on obtient, en échantillonnant la position x_0 d'émission de la particule puis la direction ω_0 dans l'espace des directions possibles sur la source selon une certaine FDP p :

$$\Phi_0 = W_e(x_0, \omega_0) \frac{L_e(x_0, -\omega_0) \cos \theta_0}{p(x_0)p(\omega_0|x_0)} \quad (3.19)$$

Généralement, le terme :

$$\Phi_0 = \frac{L_e(x_0, -\omega_0) \cos \theta_0}{p(x_0)p(\omega_0|x_0)} \quad (3.20)$$

est appelé "puissance initiale" transportée par la particule. Bien sur, le choix de la FDP est crucial et elle doit couvrir au mieux la surface de la source et la répartition de l'énergie émise.

Le problème de cette méthode est que la surface du capteur est souvent infime par rapport aux dimensions de la scène. Donc, la probabilité qu'un photon finisse sa trajectoire sur le capteur est très faible. Par conséquent, il faudrait générer un nombre immense de particules pour obtenir une image correcte alors que seul un petit nombre y contribue. Dutré [Dut96] résout ce problème grâce à un échantillonnage "explicite du capteur". Il considère que chaque pixel de l'image est un capteur indépendant. Comme dans le cas des sources lumineuses et de l'approche directe, par dualité, le capteur est le récepteur qui nous intéresse plus que toutes les autres surfaces. Donc, il suffit de viser explicitement le capteur à chaque réflexion pour accélérer la convergence vers la solution. C'est-à-dire qu'à chaque point d'impact du photon, Dutré trace un rayon vers le(s) capteur(s) et y ajoute la contribution de la particule modulée par un poids égal à la probabilité d'occurrence du phénomène. La figure 3.4 illustre cet algorithme.

De son côté, Pattanaik a plutôt une approche scène car il considère que chaque surface est en fait un capteur virtuel [PM92] captant les particules qui l'intersectent. Lors de la visualisation proprement dite, le flux incident reçu par une surface (le capteur virtuel) peut être estimé par la somme des particules qui ont intersecté cette surface. Arvo [Arv86] utilisait des surfaces texturées dont chaque texel était un capteur virtuel pouvant recevoir l'énergie transportée par les particules. L'approche du lancer de particules est de fait très générale et toutes sortes de "mesures" peuvent être envisagées selon le type de capteur utilisé.

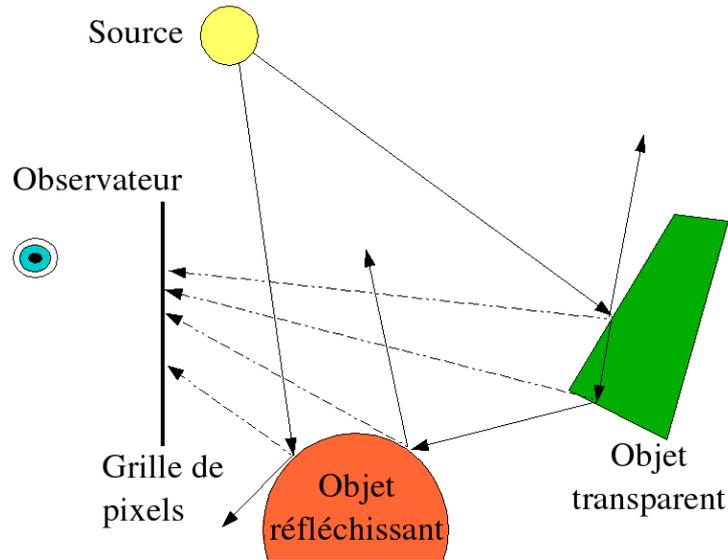


Figure 3.4. Principe du lancer de particules.

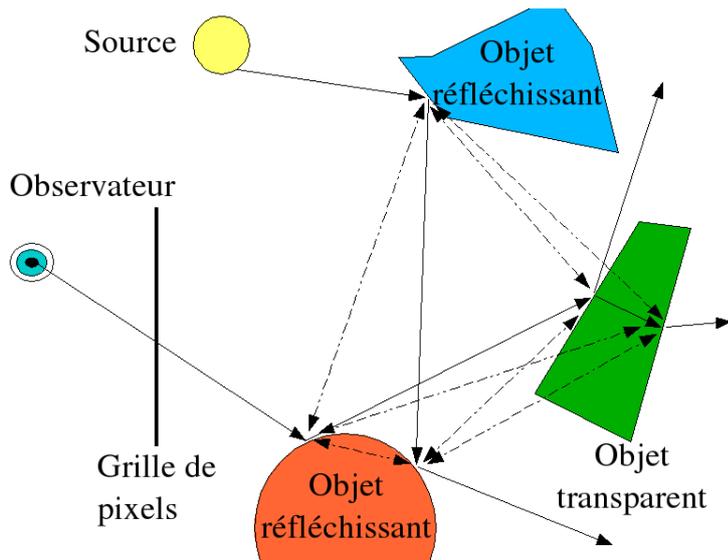


Figure 3.5. Principe du tracer de chemins bidirectionnel.

Tracer de chemins bidirectionnel Le tracer de chemins bidirectionnel [LW93, VG94] combine le tracer de chemins et le lancer de particules. Un chemin $x_0x_1\dots x_n$ tracé depuis le capteur est combiné avec un chemin $y_0y_1\dots y_{n'}$ tracé depuis les sources de lumière. L'ensemble des chemins possibles entre chacun des points x_i et y_j et dont la longueur est comprise entre 1 et $n+n'$ est construit. La valeur calculée pour chacun des chemins est ensuite ajoutée à l'estimation finale avec un poids égal à la probabilité de génération du chemin. La figure 3.5 illustre cet algorithme.

La mise en oeuvre de cet algorithme est lourde, complexe et délicate, notamment sur la notion de poids des chemins [VG97]. Le tracé de chemins bidirectionnel a été développé à l'origine pour

améliorer l'échantillonnage des chemins lumineux, néanmoins Veach prouva que l'approche classique d'échantillonnage local devait être remise en cause. Il proposa donc une nouvelle méthode, nommée *Metropolis Light Transport* [Vea97] qui, au lieu de choisir un nouveau sommet pour un chemin à partir des informations locales à la surface, effectue des perturbations globales sur des chemins générés de façon locale. Il s'agit probablement de la stratégie la plus rigoureuse existant aujourd'hui pour calculer l'illumination globale d'une scène, mais sa complexité et son coût calculatoire la rendent difficilement utilisable.

3.2.3 Illumination globale par estimation de densité

Le calcul de l'illumination globale par les méthodes précédentes reste, malgré les optimisations, très coûteux en terme de temps de calcul. Depuis le début des années 1990, de nouvelles méthodes d'illumination globales ont été développées. Elles sont basées sur l'estimation de densité pour la résolution de l'intégralité ou seulement d'une partie de l'EQR, ce qui permet une réduction des temps de calcul. Nous allons tout d'abord exposer le principe fondamental de l'estimation de densité. Puis nous décrirons son application au problème de l'illumination globale.

3.2.3.1 Principe de l'estimation de densité par noyau

La méthode d'estimation de densité par noyau est utilisée en statistique pour estimer la densité de probabilité d'un ensemble de données observées qui suivent un processus stochastique. Cet estimateur est généralement appelé non paramétrique par comparaison aux méthodes dites paramétriques qui supposent que la forme de la densité de probabilité est particulière, par exemple que c'est une gaussienne. Donc, l'estimateur \hat{f} d'une densité f à partir d'un ensemble de N points observés X_1, X_2, \dots, X_n est [Sil86] s'écrit :

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \quad (3.21)$$

avec d la dimension du domaine de f , h le paramètre de lissage et K la fonction noyau. La fonction noyau K doit être unitaire, c'est à dire $\int K(x)dx = 1$. Elle est aussi généralement choisie symétrique, ce qui garantit que l'estimation $\hat{f}(x)$ est toujours une densité. Enfin, elle a souvent un support compact, ce qui est intéressant lors du calcul de l'estimateur. Dans la littérature statistique, cet estimateur est aussi mis parfois sous une forme plus simple et plus générale :

$$\hat{f}(x) = \sum_{i=1}^N \alpha_i K_h(x - X_i) \quad (3.22)$$

avec $K_h(x) = \frac{1}{h^d} K(\frac{x}{h})$ et α_i le poids associé à la donnée X_i généralement égal à $\frac{1}{N}$. Un poids différent peut être associé à chaque donnée à condition que la somme de ces poids soit égale à 1.

L'intérêt de cet estimateur est simple : c'est une moyenne pondérée par la fonction noyau centrée sur le point d'estimation. Cet estimateur peut aussi être vu comme la somme des n fonctions noyaux centrées sur les données observées comme illustré sur la figure 3.6.

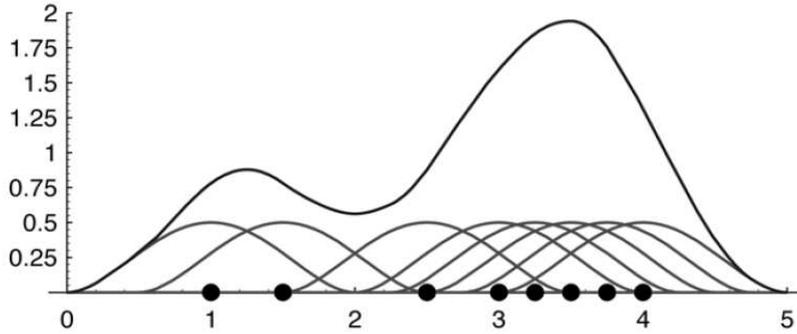


Figure 3.6. Estimation de densité par noyau.

Cet estimateur est stochastique puisqu'il est basé sur des observations aléatoires. L'erreur de cet estimateur est caractérisée par la variance et le biais. La variance se présente sous la forme de bruit sur la densité reconstruite, et le biais se présente sous la forme d'une densité estimée trop lissée. Mathématiquement, le biais de l'estimateur \hat{f} est égal à :

$$\text{Biais}(\hat{f}(x)) = E(\hat{f}(x)) - f(x) \quad (3.23)$$

Cet estimateur est biaisé car son espérance est différente de $f(x)$. En effet, l'espérance de $\hat{f}(x)$ est égale à [Sil86] :

$$E(\hat{f}(x)) = \int K_h(x - y) f(y) dy \quad (3.24)$$

L'espérance ne converge donc pas vers la vraie densité, mais vers cette densité convoluée avec le noyau. Cette convolution provoque effectivement un lissage de la densité estimée.

En jouant sur les différents paramètres de l'estimation de densité (n , le nombre de données aléatoires observées, K la fonction noyau et h le paramètre de lissage), on peut jouer sur la vitesse de convergence de l'estimation et sur la précision obtenue. Plus de précisions sur l'influence de ces paramètres peuvent être trouvées dans [Lav03].

3.2.3.2 Application à l'illumination globale

En 1990, Heckbert [Hec90] étend la méthode du lancer de particules d'Arvo [Arv86]. Il introduit la construction d'une texture adaptative pour stocker l'illumination. Il applique ainsi aux méthodes de lancer de rayons le principe des méthodes de radiosit ,   savoir l'utilisation d'une m thode en deux passes (i.e. pr calcul de l'illumination, puis visualisation du maillage). Il est  galement le premier   faire le parall le entre les m thodes de trac  de particules et l'estimation de densit . En 1991, Chen et al. [CW91] proposent une m thode bas e sur l'estimation de densit  par noyau sp cifiquement pour l' valuation des caustiques. Leur approche a  t   tendue   tous les effets d'illumination globale par Shirley [SWH⁺95]. Collins [Col94] propose  galement une m thode d'estimation de densit  efficace, mais elle ne prend pas en compte les diffusions multiples. En 1995, Jensen [Jen95] introduit la m thode des *cartes de photons* (en anglais *Photon mapping*) qui est devenue une r f rence parmi les m thodes d'illumination globale. Celle-ci a ensuite  t  sujette   de nombreuses  volutions, la plupart  tant d crite dans [Jen01]. En parall le, Myzskowski en 1997 [Mys97], Walter en 1998 [Wal98] et Volevich en 2000 [VMKE00] ont  galement propos  des m thodes similaires.

Nous allons tout d'abord formaliser l'utilisation de l'estimation de densit  par les m thodes d'illumination globale, puis nous les comparerons. Enfin, nous exposerons les am liorations qui leur ont  t  appliqu es.

Analyse formelle Toutes les m thodes de calcul de l'illumination globale par estimation de densit  fonctionnent sur le m me principe. L'algorithme est g n ralement constitu  de deux phases :

- propagation de l'illumination depuis les sources lumineuses,
- reconstruction de l'illumination par estimation de densit .

La propagation de l'illumination est toujours effectu e par un trac  de particules similaire au lancer de particules. La diff rence entre les m thodes r side dans le stockage et l'exploitation des impacts des photons.

Nous avons vu qu'en lancer de particules, le flux incident Φ_S re u par une surface S peut  tre estim  par la somme de la puissance des N_S photons qui ont eu une intersection avec cette surface. Cela est r alis  en prenant comme fonction de r ponse W_e une fonction qui est  gale   1 si la position du photon appartient   cette surface S , et 0 sinon. Soit N le nombre de photons,

Φ l'énergie moyenne des N photons, et ϕ_p l'énergie du p -ième photon :

$$\Phi_S = \sum_{p=1}^{N_S} \phi_p \approx N_S \phi \quad (3.25)$$

La probabilité qu'un photon P_p appartienne à la surface S est alors :

$$P(P_p \in S) = \frac{N_S}{N} = \frac{N_S \phi}{N \phi} = \frac{\Phi_S}{N \phi} \quad (3.26)$$

On peut en déduire que la densité de probabilité $p(x)$ de la présence des photons dans la scène :

$$p(x) = \frac{dP}{dx} = \frac{d\Phi}{N \phi} \frac{1}{dx} = \frac{E_i(x)}{N \phi} \quad (3.27)$$

Cette formulation indique que la densité de probabilité des photons est proportionnelle à l'éclairement incident. Si l'on estime la densité $p(x)$, on obtient également une estimation de cet éclairement. En combinant l'estimateur de densité par noyau et l'équation précédente, on obtient un estimateur de l'éclairement incident, à savoir :

$$E_i(x) = \frac{N \phi}{N} \sum_{p=1}^N K_h(x - X_p) = \phi \sum_{p=1}^N K_h(x - X_p) \quad (3.28)$$

En levant la restriction sur l'égalité de l'énergie des photons, on obtient :

$$p(x) = \sum_{i=p}^N \frac{\phi_p}{N \phi} K_h(x - X_p) \quad (3.29)$$

Cette équation est tout à fait similaire à l'équation de l'estimation de densité par noyau avec un poids pour chaque donnée aléatoire (équation 3.22). En effet, la condition $\sum_{p=1}^N \frac{\phi_p}{N \phi} = 1$ est bien respectée puisque ϕ est la moyenne de l'énergie des photons. L'équation pour estimer l'éclairement est donc :

$$E_i(x) = \sum_{p=1}^N \phi_p K_h(x - X_p) \quad (3.30)$$

Nous avons donc obtenu un estimateur de l'éclairement incident. Néanmoins, l'œil n'est pas sensible à l'éclairement mais à la luminance, et plus particulièrement la luminance réfléchie par les surfaces vers l'œil. Reprenons l'expression de la BRDF :

$$f_r(x, \omega_i, \omega_r) = \frac{dL_r(x, \omega_r)}{dE_i(x, \omega_i)} = \frac{dL_r(x, \omega_r)}{L_i(x, \omega_i) \cos \theta_i d\omega_i} \quad (3.31)$$

Ainsi, on peut exprimer la variation de la luminance réfléchie en x dans la direction de l'œil ω_r en fonction de la variation de l'éclairement incident et de la BRDF :

$$dL_r(x, \omega_r) = f_r(x, \omega_i, \omega_r) dE_i(x, \omega_i) \quad (3.32)$$

En remplaçant l'éclairement incident par son estimateur (équation 3.30), on obtient :

$$dL_r(x, \omega_r) = f_r(x, \omega_i, \omega_r) d\left(\sum_{p=1}^N \phi_p K_h(x - X_p)\right) \quad (3.33)$$

On intègre sur l'espace des directions incidentes :

$$L_r(x, \omega_r) = \int_{\Omega=4\pi} f_r(x, \omega_i, \omega_r) \sum_{p=1}^N \phi_p K_h(x - X_p) d\omega_i \quad (3.34)$$

Remarquons ici que l'énergie portée par les photons est directionnelle. En effet, elle est donnée pour la direction incidente du photon ω_i et pour la position du photon X_p . Etant donné que les seules données que nous possédons ici sont les photons, l'espace des directions incidentes se résume aux directions des photons. Pour chaque photon, l'intégrale est donc non nulle uniquement pour sa direction. Ce qui équivaut à écrire que $\phi_p = \phi(X_p, \omega_i)$. L'équation précédente devient donc :

$$L_r(x, \omega_r) = \sum_{p=1}^N f_r(x, \omega_i, \omega_r) \phi_p K_h(x - X_p) \quad (3.35)$$

Finalement, on a une expression de l'estimateur de la luminance réfléchie vers l'œil en un point x . Cet estimateur possède les mêmes propriétés que l'estimateur par noyau, en particulier le compromis biais/variance contrôlé par le paramètre de lissage. Les processus stochastiques ont l'avantage d'être plutôt simples à mettre en œuvre et de donner un résultat correct, assez rapidement, bien que bruité.

Grâce à l'estimateur d'éclairement (équation 3.30), on peut reconstruire l'illumination pour les points d'estimation d'un maillage, auquel cas, la luminance est calculée lors de la visualisation de ce dernier. L'estimateur de luminance (équation 3.35) est plus direct, car il permet de reconstruire directement la luminance à partir des photons stockés. Dans ce cas, les points d'estimation sont directement les pixels de l'image. L'intérêt de ces deux expressions équivalentes est d'unifier les méthodes d'estimation de densité, qu'elles soient basées scène ou image.

Comparaison des différentes méthodes Comme nous l'avons indiqué précédemment, toutes les méthodes d'estimation se basent sur le principe de précalcul et stockage de l'illumination, puis sur son exploitation lors de la visualisation. La première différence réside dans le type de stockage utilisé. Heckbert [Hec90] choisit de stocker l'illumination diffuse dans des textures adaptatives (qu'il nomme *rexes*). Ces textures sont subdivisées en fonction du gradient d'illumination à la manière d'un *quad-tree*. Il préfère ces textures aux maillages utilisés par les méthodes de radiosité

afin de se détacher de la complexité géométrique de la scène. Partant du principe que l'illumination est intrinsèque aux matériaux, il utilise des textures plaquées sur les objets de la scène. Le nombre de textures peut donc être limitant lorsque l'on souhaite traiter des scènes complexes, et l'on s'aperçoit que la méthode reste malgré tout dépendante de la complexité géométrique. Collins [Col94] préfère utiliser un maillage d'illumination. Celui-ci est construit adaptativement lors de la propagation des photons. Il est composé des impacts des photons sur les surfaces de la scène. Ce maillage est uniquement dépendant du nombre de photons tracés depuis les sources. Néanmoins, la précision de la simulation augmente avec ce nombre. Bien qu'indépendante de la complexité géométrique de la scène, la méthode peut être mise en échec si l'on veut augmenter le réalisme. L'approche de Shirley [SWH⁺95] ressemble aux deux précédentes. La différence réside sur l'observation de Lischinski *et al.* [LTG93] que le maillage de visualisation n'est pas forcément le maillage utilisé pour le calcul. Ainsi, Shirley retarde l'exploitation des photons jusqu'à la fin de la phase de propagation en les stockant et les associant avec les surfaces de la scène qu'ils ont frappées. Ces impacts constituent un maillage d'illumination similaire à celui de Collins. L'estimation de densité par noyau est alors utilisée pour calculer l'éclairement de ce maillage. Afin de réduire sa complexité, ce dernier subit enfin une phase de simplification basée sur le gradient d'illumination. Cette approche a été reprise par Myzskowski [Mys97] et Walter [Wal98] qui reconstruisent l'illumination sur les surfaces de la scène. Leurs évolutions portant essentiellement sur la sélection adaptative de la taille du support du noyau (paramètre h).

Cependant, la reconstruction de la luminance sur chaque surface de la scène utilise uniquement les photons qui l'ont frappée. Le nombre de particules étant limité, la probabilité d'avoir suffisamment d'information sur les petites faces est faible ce qui peut générer des artefacts visuels. Jensen [Jen95] propose la méthode des *cartes de photons* qui résout à la fois le problème de dépendance vis-à-vis de la complexité de la scène et le problème de sous-estimation des petites faces. L'idée est de stocker les photons dans une structure de données indépendante de la géométrie qui permet une recherche rapide des photons. En effet, les points d'estimation x ne sont plus les points du maillage, mais l'intersection des rayons primaires avec les surfaces de la scène. L'estimation de densité est effectuée en sélectionnant les k plus proches photons de x contenus dans une sphère de rayon r (paramètre de lissage pour l'estimation) pour reconstruire le flux incident reçu par x . Ce type d'estimation est un cas particulier de l'estimation de densité par noyau. En fait, Jensen sélectionne les photons participant à l'estimation s'ils appartiennent à une sphère de rayon égal au paramètre de lissage centrée sur ce point. Par contre, dans la for-

mule de l'estimation, le noyau est supposé comme ayant un support 2D, c'est-à-dire un disque. L'inconvénient de cette méthode est que des photons appartenant à une surface très différente peuvent être sélectionnés et ainsi augmenter le biais sur l'estimateur. Pour éviter cela, Volevich [VMKE00] utilise, entre autre, différentes heuristiques pour sélectionner ses "photons". Par exemple, l'angle entre la normale à la position du photon et la normale au point d'estimation est calculé, et le photon est rejeté si cet angle est trop grand.

Toutes les méthodes que nous avons citées dans ce paragraphe sont des méthodes dites d'estimation locale. Le principe est de chercher pour chaque point d'estimation l'ensemble des échantillons (photons) qui contribuent à l'éclairage de ce point. L'inconvénient majeur de cette méthode est que l'on doit rechercher les échantillons contribuant dans l'ensemble des échantillons. Cela ne pénalise pas la méthode dans le cas où le nombre d'échantillons est raisonnable (e.g. photons rangés par surface [SWH⁺95, Mys97, Wal98]). Cependant nous avons vu que ces méthodes sont dépendantes de la complexité de la scène. Dans le cas de [Jen95], par contre, tous les photons sont stockés dans une unique structure de donnée. La recherche des échantillons contribuant est donc effectuée sur un grand nombre de photons. Les mêmes photons peuvent être utilisés pour plusieurs points d'estimation, ce qui cause des recherches redondantes et donc des temps de calcul plus élevés. Partant de l'observation que le nombre de points d'estimation (intersection des rayons primaires avec la scène) est moins important que le nombre de photons, Lavignotte [Lav03] propose une méthode d'estimation globale appelée *écrasement de photons*. En effet, il est alors plus intéressant de rechercher pour chaque photon les points d'estimation auxquels il contribue. Les photons peuvent être traités séquentiellement ce qui permet d'éviter les recherches redondantes coûteuses. On effectue malgré tout des recherches redondantes sur les points d'estimation, mais c'est un moindre mal car le coût de chaque recherche est moins élevé. La figure 3.7 montre la différence entre les estimations locale et globale. On parle aussi d'approches directe et duale de l'estimation de densité.

Nous décrirons plus précisément les méthodes des cartes de photons [Jen95] et de l'écrasement de photons [Lav03] respectivement dans les chapitres 4 et 5 car elles constituent la base de nos travaux. Ces méthodes sont en effet celles qui proposent le meilleur compromis entre temps de rendu et qualité de simulation. Néanmoins, le stockage des photons nécessaire dans ces méthodes engendre une consommation mémoire importante, limitant le nombre de photons et par conséquent la précision de la méthode. C'est pourquoi nous proposons dans le chapitre 4 une méthode multi-passes permettant de lever la limitation mémoire.

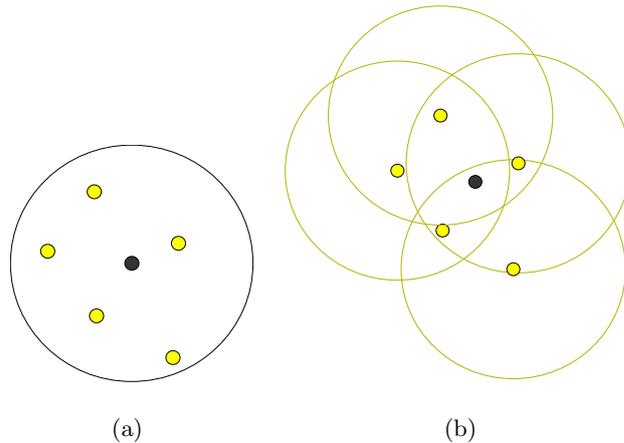


Figure 3.7. Estimation de densité. (a) Locale : on recherche les photons (en jaune) qui contribuent aux points d'estimation (en noir). (b) Globale : on recherche les points d'estimation auxquels le photon contribue.

Améliorations Ces méthodes, et notamment la méthode des cartes de photons, ont bénéficié de nombreuses améliorations afin d'accélérer les temps de calcul, de réduire les artefacts...

Correction du biais sur les bords Nous avons vu que l'estimation par noyau avait des difficultés sur les bords du domaine de définition. Shirley et al. [SWH⁺95] ont proposé de compléter le bord du domaine par symétrie. Cependant cette méthode n'est pas applicable si la frontière du domaine est complexe (ce qui est souvent le cas pour des scènes complexes). Walter [Wal98] propose, quant à lui, une approche basée sur la méthode de régression polynomiale. Il a proposé deux nouvelles évaluations de l'estimation sur les bords : la méthode locale constante et la méthode locale linéaire. Cette dernière supprime totalement le biais sur les bords si la densité sur les bords a un gradient constant et la méthode constante supprime ce biais si la densité est localement constante sur les bords. Cependant, la méthode linéaire est beaucoup plus complexe et coûteuse. Lavignotte [Lav03] propose une approche géométrique de correction du biais sur les bords. Il introduit la notion de "surfaces fantômes". En fait, la géométrie est étendue sur une largeur égale au paramètre de lissage de la fonction noyau. Lors du lancer de photons, les impacts sur les surfaces fantômes sont stockés et les photons sont réémis comme s'ils n'avaient rien rencontré. En effet, les photons passant près des surfaces participent bien au flux lumineux. Ainsi, lors du rendu, la densité de photons au bord permet d'estimer correctement le flux lumineux.

Importance visuelle Les méthodes précédentes calculent l'illumination indirecte indépendamment du point de vue. C'est un avantage, car il n'est pas nécessaire de recalculer l'éclairage indirect lorsque l'on navigue dans la scène. Cependant, dans les cas où seul un point de vue nous intéresse, il est possible d'optimiser le calcul en limitant la précision du calcul sur les zones peu intéressantes pour l'observateur. Ainsi, Suykens [SW00] propose d'effectuer une passe préliminaire où des "importons" sont lancés depuis le capteur. Les importons sont similaires à des photons et sont déposés sur les surfaces importantes pour l'observateur. Ensuite, lors de la passe de lancer de photons, la densité d'importons est utilisée pour savoir si la densité de photons sur une surface est suffisante : dans ce cas, le photon n'est pas stocké et son énergie est répartie sur les photons déjà stockés les plus proches.

Utilisation de plusieurs petites cartes de photons Larsen [LC03] propose une amélioration de la méthode de Jensen [Jen95]. Au lieu d'utiliser une seule carte de photons, il stocke les photons dans des cartes de plus petite taille. Les différentes cartes sont construites en fonction de critères géométriques. Par exemple, un critère sur la variation de la normale d'une face à l'autre. Cette méthode présente deux avantages. Le premier est que l'utilisation de plusieurs petites cartes de photons est plus rapide que l'utilisation d'une grande. Deuxièmement, la méthode corrige le biais sur les bords. En effet, les photons stockés sur deux faces d'un coin sont dans des cartes différentes. On ne sélectionne donc pas de "mauvais" photons pour l'estimation de densité. Cependant, cette méthode n'offre pas de bons résultats pour des scènes complexes.

3.3 Milieux participants

Les méthodes décrites précédemment prennent toutes pour hypothèse que la lumière se propage dans le vide. Elles négligent alors la participation des milieux, ce qui introduit nécessairement un biais par rapport à la solution réelle. De nombreux travaux ont été effectués dans le but de pallier ce manque. Nous verrons que les méthodes proposées sont loin d'être générales et font quasiment toutes des hypothèses importantes sur les propriétés des milieux (e.g. isotropie, homogénéité...). Nous mettons ici en balance les méthodes dites déterministes et les méthodes stochastiques afin de déterminer la méthode la plus performante pour simuler des milieux participants complexes.

3.3.1 Méthodes déterministes

3.3.1.1 Méthodes de radiosité

Méthode zonale [RT87] La méthode zonale est une extension de la méthode de radiosité qui inclut un milieu participant isotrope, modélisé par des voxels. La radiosité du voxel est définie afin de prendre en compte uniquement l'émission propre plus l'énergie diffusée. Des facteurs de forme sont définis pour les échanges entre des volumes, et entre un volume et une surface. Les facteurs de forme entre surfaces sont redéfinis pour inclure un terme de "transmittance". Ces facteurs sont calculés en étendant la méthode de l'hémicube. Un système de s (pour surfaces - éléments) plus v (pour volumes - voxels) équations est construit puis résolu par la méthode itérative de Gauss-Seidel. L'application directe de la méthode zonale a un coût prohibitif. Dans un cube de n^3 voxels, il y a n^6 facteurs de forme. En les approchant par l'intégrale 1D sur le centre de chaque paire de voxels en $O(n)$, le calcul de tous les facteurs de forme est en $O(n^7)$. La cohérence entre les facteurs de forme a été exploitée pour les calculer à moindre coût [AM96].

Radiosité hiérarchique Sillion [Sil95] présente un algorithme de radiosité hiérarchique adapté pour traiter des volumes isotropes. Afin de représenter les échanges à l'intérieur du volume, on introduit un "auto-lien" (lien entre le volume et lui-même). Ce lien est subdivisé différemment des liens entre les différents éléments, puisque chaque lien fils doit inclure un auto-lien en plus des liens usuels entre chaque paire de fils. De plus, pour éviter le coût quadratique de la phase de lien initiale de la méthode hiérarchique classique, qui peut être prohibitif pour des scènes complexes, le transfert d'énergie entre des groupes d'objets est autorisé. Ces groupes composent des objets abstraits qui échangent l'énergie comme un tout. Une hiérarchie est créée au-dessus du niveau des surfaces, et la phase initiale de lien est réduite à la création d'un seul auto-lien entre le haut de la hiérarchie et elle-même, qui représente les interactions à l'intérieur du volume global contenant la scène. Une fois que le lien initial a été raffiné par une procédure récursive, les étapes de rééquilibrage de l'énergie (*gathering* et *Push-Pull*) sont effectuées jusqu'à ce qu'il n'y ait plus de changement significatif de la radiosité entre tous les éléments. On doit faire attention lors de l'application de ces étapes quand on traite des milieux non homogènes et des surfaces texturées. Le raffinement des liens est effectué par seuillage du transfert de radiosité.

3.3.1.2 Harmoniques sphériques

Kajiya [KH84] présente deux méthodes. La première ne traite que la diffusion simple dans les milieux participants. La seconde traite la diffusion multiple. La luminance est exprimée dans une base d'harmoniques sphériques. Un système d'équations aux dérivées partielles (SEDP) est alors construit pour les coefficients des harmoniques sphériques. Le SEDP est posé et résolu par relaxation. Seules les fonctions de phase isotrope et de Rayleigh sont considérées, et le développement en harmoniques sphériques est tronqué après le quatrième coefficient (car "seules les quelques premières harmoniques sphériques sont nécessaires pour avoir une image convaincante", mais le coût de la méthode dépend évidemment du nombre de coefficients). Les effets entre les surfaces et les volumes ne sont pas pris en compte. La méthode proposée par Bhat [BT92] traite les effets entre les surfaces et les volumes qui manquent dans [KH84]. C'est une extension de la méthode zonale, dans laquelle l'hypothèse d'isotropie est éliminée (grâce à une représentation de la fonction de phase et de la luminance utilisant les harmoniques sphériques) et où les surfaces restent lambertiennes (parfaitement diffuses). La fonction de phase est approchée par les M premiers termes de son développement en harmoniques sphériques (approximation de la diffusion de Mie). Il faut noter qu'un grand nombre de facteurs de forme doivent être calculés, en prenant en compte les harmoniques sphériques. Ils sont calculés par la technique étendue de l'hémicube. Finalement, un système de Mv équations pour v volumes plus s équations pour s surfaces est établi et résolu en utilisant la méthode itérative de Gauss-Seidel. L'application directe de cette méthode est impraticable à cause de son coût prohibitif : dans une grille régulière de $v = n^3$ voxels, le coût de calcul des facteurs de forme est de $O(n^7 + M^2n^6)$.

3.3.1.3 Ordonnées discrètes

Une autre possibilité pour prendre en compte les fonctions directionnelles (phase) est d'utiliser les ordonnées discrètes, c'est-à-dire une discrétisation de l'angle solide total (4π) en angles solides élémentaires. Ceux-ci représentent des directions particulières, et on suppose que, pour des éléments de volumes suffisamment petits, les propriétés sont constantes pour chaque direction dans chaque volume. Le problème principal des ordonnées discrètes vient du fait que l'énergie est propagée dans des directions discrètes au lieu de tout l'angle solide discrétisé.

Interactions locales Patmore [Pat93] formule la solution locale de l'équation du transfert radiatif pour le modèle directionnel discret résultant de la subdivision du volume (résultant en

un réseau cubique) et l'espace angulaire (utilisant en pratique 6 ou 26 directions). Le milieu participant considéré n'est pas émissif, puisque l'objectif est de rendre des nuages. Une solution globale de l'ETR est obtenue en trouvant itérativement des solutions locales (relatives aux points du réseau cubique). L'énergie non émise du point considéré est mise à jour en fonction de la solution locale. Une nouvelle solution locale est calculée pour un point adjacent du réseau et dans la direction de la plus forte énergie non émise du point précédent. On suit donc les chemins choisis par importance, jusqu'à ce que l'énergie non émise soit au-dessous d'un seuil ou que le chemin sorte du volume. Cette méthode calcule directement les luminances sortant du volume. Aucune intégration de la luminance des sources n'est nécessaire dans la passe de visualisation. La méthode de Languénou [LBC94], suit une approche de raffinement progressif. La méthode habituelle de tirages pour les surfaces est étendue pour prendre en compte la transmittance à travers le milieu. Les radiosités des frontières du média sont calculées en propageant la luminance (précédemment calculée) le long de toutes les ordonnées discrètes et en effectuant autant d'itérations que nécessaire pour converger. Chaque itération consiste en une boucle pour chaque direction, dans laquelle un balayage complet de la grille de voxels est effectué pour propager l'énergie accumulée à travers les voxels voisins, en commençant par un voxel de la frontière convenant (en relation avec la direction considérée), où $O(Mv)$ est le coût par itération. Finalement, la luminance des faces de la frontière du milieu est traitée, en utilisant l'interpolation hémisphérique. L'ensemble du processus est répété jusqu'à convergence. La passe de visualisation calcule la luminance des pixels en utilisant la luminance des voxels.

Interactions globales La méthode proposée par Max [Max94] est destinée à rendre des nuages. Le calcul des M^2v^2 facteurs de forme de la formulation en éléments finis est évité en approchant leurs effets par l'énergie propagée sur l'ensemble des angles solides élémentaires. Pour chacun d'entre eux, la propagation est effectuée en distribuant le flux aux voisins simultanément pour tous les voxels appartenant à une couche, avec un temps de $O(v \log(n))$, pour v^2 interactions. Le problème dû à la discrétisation de l'espace des directions est réduit car l'énergie est propagée dans l'angle solide élémentaire, et non plus simplement dans une seule direction. L'atténuation entre deux voxels n'est pas accumulée le long de la ligne droite les reliant, mais le long d'un ensemble de chemins de propagation possibles. Les événements de diffusion multiple produite avec un seul élément récepteur sont pris en compte. Comme le temps pour diffuser le flux réfléchi par un voxel dans les angles solides élémentaires est $O(Mv)$, le coût final par

itération est $O(Mv \log(n) + M^2v)$. Quand le nombre d'itérations requis pour converger est petit comparé à v , cette méthode est meilleure que le calcul de l'ensemble des facteurs de forme et de résolution du système résultant.

3.3.1.4 Représentation implicite

La distribution directionnelle de la luminance peut être représentée implicitement par des modèles de diffusion [NDN96] ou par une équation de diffusion [Sta95b, Sta95a].

Filtre 3D Nishita [NDN96] propose une méthode pour rendre des nuages qui prend en compte la diffusion multiple et la lumière du ciel (la lumière atteignant le nuage due à la diffusion de l'atmosphère plus la lumière réfléchiée par la surface du sol). La luminance d'un nuage atteignant l'œil est calculée en utilisant la lumière du soleil issue de diffusions multiples plus la lumière du ciel diffusée une seule fois par les particules du nuage. De toutes les diffusions de la lumière du soleil dans le nuage, seules les trois premiers ordres de diffusion sont considérés. On calcule séparément la diffusion simple. Pour la contribution des deuxième et troisième ordres de diffusion à la luminance de la direction de l'œil, l'espace contenant le nuage est subdivisé en voxels, avec la direction de l'œil comme axe principal. Au lieu de calculer les facteurs de forme entre chaque paire de voxels, un petit espace, contenant la densité moyenne du nuage, est mis en place. Les rapports de contribution entre la luminance du voxel du centre (dans la direction de l'œil) et les autres voxels sont calculés, en prenant en compte la lumière du soleil. Ceci est le modèle des rapports de contribution (*contribution-ratio pattern*) ou filtre 3D. Comme la diffusion dans les nuages est surtout "en avant", la plus grande partie de l'énergie diffusée en un point est comprise dans un angle solide relativement petit. En utilisant cette observation, il est possible de calculer plus rapidement les facteurs de forme étendus, en concentrant l'effort sur les voxels qui contribuent effectivement au voxel central, pour les chemins qui ont une ou deux diffusions (et en utilisant une méthode stochastique pour choisir ces voxels). Le filtre est appliqué aux voxels dans tout l'espace en stockant pour chaque voxel la lumière diffusée due aux seconde et troisième diffusion dans la direction de l'œil.

Diffusion Stam [Sta95b, Sta95a], résout l'illumination globale par raffinements progressifs, en utilisant les opérations d'émission entre éléments de surface (patches) et entre patches et blobs (qui modèlent le milieu). L'émission entre blobs (qui peut être très coûteuse si le nombre de blobs v est

grand) est évitée en utilisant un ensemble de v équations linéaires qui représentent une équation de diffusion. Celle-ci est obtenue en faisant une "approximation de diffusion" de la luminance source (due à la diffusion de la luminance du milieu), c'est-à-dire qu'elle est caractérisée par seulement deux fonctions : $J_m(\omega) = J_0 + J_1\omega$. Résoudre le système linéaire permet le calcul des coefficients J_0 et J_1 pour chaque blob, et donc la diffusion multiple entre les blobs. Quand v n'est pas trop grand ($v < 1000$), le système peut être résolu avec une décomposition LU directe. Pour des systèmes plus larges, on peut résoudre le système par relaxation, néanmoins la convergence n'est pas garantie (il faut noter que de bons résultats sont déjà obtenus avec peu de blobs). La méthode proposée utilise beaucoup moins de mémoire et de temps de calcul qu'une méthode utilisant une grille régulière. Cette méthode étant progressive, pour le traitement de scènes complexes composées d'un grand nombre de surfaces, le coût de l'émission progressive de l'énergie depuis les patchs est très coûteux. Une approche hiérarchique deviendrait nécessaire dans ce cas.

3.3.2 Méthodes stochastiques

Les méthodes stochastiques d'illumination globale tracent des rayons dans l'environnement. Les points d'interactions avec le milieu qui limitent les rayons peuvent être obtenus en utilisant une distance constante [BLS93, BLS94] ou une fonction de densité cumulative d'échantillonnage [PM93, LW96].

3.3.2.1 Echantillonnage à distance constante

Blasi [BLS93, BLS94] décrit des méthodes qui traitent les milieux participants qui utilisent une simulation du modèle corpusculaire de la lumière (Monte Carlo light tracing). La première méthode prend en compte un milieu participant seul, la seconde des scènes mixtes (milieu plus objets). Les deux méthodes simulent la diffusion multiple dans le milieu participant. La fonction de phase utilisée est celle de Schlick [BLS93], spécialement définie de façon à ce que l'échantillonnage par importance soit peu coûteux, tout en maintenant la possibilité d'approcher d'autres fonctions de phase. Dans [BLS93], on utilise une approximation de la diffusion de Mie comme une combinaison de composantes isotropes et en avant. Lors des événements de diffusion, la direction de diffusion est donnée par un échantillonnage par importance de la composante de diffusion, et la partie isotrope est stockée dans le voxel. Cette partie isotrope du voxel n'est pas considérée pour l'illumination des autres voxels. Comme la composante directionnelle est plus importante que la

composante isotrope, on attend que l'erreur commise ne soit pas significative. Une stratégie de raffinements progressifs peut être utilisée quand l'énergie isotrope devient trop importante. Les faisceaux progressent par pas de longueur constante. A chaque point d'interaction, un processus d'échantillonnage est utilisé pour décider s'il y a diffusion en ce point. L'absorption est prise en compte le long de l'ensemble du chemin du faisceau : son flux est diminué à chaque pas par la transmittance due à l'absorption le long de la distance δ . Dans [BLS94], une technique progressive est utilisée pour rendre des scènes mixtes. Les surfaces sont classées comme "diffuses" ou "spéculaires" en fonction d'un seuil. Pendant la passe d'illumination, quand un faisceau frappe une surface diffuse, son énergie est stockée au point d'intersection (et le faisceau s'arrête là), alors que s'il frappe une surface spéculaire, il est réfléchi en utilisant un échantillonnage par importance. A l'intérieur du milieu, les faisceaux progressent comme expliqué ci-dessus. L'énergie du faisceau est enregistrée (au bord du voxel) lorsque le faisceau sort du milieu(s'il sort). Ce stockage implique que le nombre de rayons qui traversent le milieu doit être plus grand que si on effectuait un stockage par voxel pour avoir un échantillonnage précis de l'énergie quittant le volume.

3.3.2.2 Echantillonnage à distance aléatoire

Lancer de particules Le lancer de rayons depuis les sources, proposé par Pattanaik [PM93], utilise une étape d'échantillonnage pour trouver les points d'interaction (absorption ou diffusion) des faisceaux à l'intérieur du volume. Cet échantillonnage considère l'expression $1 - \exp(-\int_0^S \sigma_t(u)du)$ comme une fonction de distribution cumulative, où S est la distance parcourue par le faisceau. En ces points, avec la méthode d'Absorption Simple, une autre étape d'échantillonnage est effectuée pour décider si l'interaction est une absorption ou une diffusion. D'un autre côté, avec la méthode de Suppression d'Absorption, le rayon est toujours diffusé mais son flux est diminué à chaque interaction. Différentes méthodes de réduction de la variance sont proposées : l'interaction forcée d'un faisceau avec chaque voxel (Suppression d'Absorption), et la méthode de la divergence des particules (dans laquelle le faisceau réfléchi est découpé en petits faisceaux au point de diffusion). Le format de stockage présenté convient bien pour la diffusion isotrope, mais elle peut être changée pour traiter les diffusions anisotropes.

Lancer de rayons bidirectionnel La méthode du lancer de rayons bidirectionnel a été étendue aux milieux participants dans [LW96]. Le principe de la méthode reste le même que dans

le cas des surfaces, mis à part que les chemins aléatoires sont tracés en calculant les interactions avec le milieu comme dans la méthode d'absorption simple de [PM93]. Pour le calcul de la direction de diffusion, la fonction de phase de Schlick est utilisée.

Metropolis light transport Pauly [PKK00] propose d'étendre la méthode de Veach et Guibas [VG97] à la prise en compte de milieux participants. Afin d'évaluer l'intégrale du rendu, il est nécessaire de construire les chemins en respectant une fonction de densité de probabilité appropriée (FDP). La génération des chemins est divisée en une séquence d'évènements de diffusion et de propagation. Un évènement de diffusion choisit une direction en un point en échantillonnant la fonction de phase. Un évènement de propagation détermine le prochain point d'interaction dans une direction de propagation à partir du point courant. Cela est fait en calculant la distance d'interaction moyenne dans un milieu participant. La FDP du chemin entier est alors simplement le produit de toutes les FDP des diffusions et des propagations qui composent le chemin, vu qu'elles sont indépendantes les unes des autres. Des mutations des chemins, similaires à la méthode classique, sont ensuite effectuées pour le rendu de la scène. Cette méthode peut prendre en compte les milieux anisotropes et/ou non homogènes et la diffusion multiple. Elle conserve les avantages de la méthode originale à savoir la prise en compte de scènes complexes sans discrétisation de la scène ou de l'espace. C'est une méthode non biaisée. Le contrôle des mutations permet d'explorer l'espace localement, ce qui permet de réduire le coût d'échantillonnage. Si un chemin important est trouvé, les chemins proches sont aussi échantillonnés. Les mutations peuvent être modifiées facilement pour trouver des chemins qui conservent une propriété particulière (ou plusieurs).

Cartes de photons [Jen01] Jensen propose une évolution de sa méthode pour prendre en compte les milieux participants. Le principe de la méthode reste le même. Une première passe propage les photons depuis les sources lumineuses, puis une deuxième passe permet de reconstruire la luminance par estimation de densité. La méthode conserve tous les avantages vus dans la section 3.2.3.2.

L'évolution concerne les deux passes de la méthode. Le milieu participant est introduit sous la forme d'une grille régulière 3D. Chaque voxel de la grille contient les données radiatives nécessaires à l'évaluation de l'équation volumique du transfert radiatif (cf. 2.3). Les propriétés du milieu sont supposées constantes sur le voxel. Lors de la première passe, les photons sont émis

dans la scène. Lorsque le rayon rencontre le volume englobant du milieu, la distance d'interaction est tirée aléatoirement :

$$d = -\frac{\ln(\xi)}{\sigma_t}$$

où ξ est un nombre aléatoire tiré uniformément entre 0 et 1, dans le cas d'un milieu homogène. Pour un milieu non homogène, le coefficient d'extinction est remplacé par l'épaisseur optique mesurée sur la distance moyenne tirée précédemment. Lors de la phase de rendu, les rayons qui traversent le milieu participant sont évalués par *Ray-Marching*. Cela consiste à diviser le rayon en petits segments (de longueur constante ou non) sur lequel on peut considérer que les propriétés du milieu sont constantes. On peut ainsi calculer la contribution de chacun des segments et donc l'énergie qui sort du milieu dans la direction du rayon.

Cette méthode sera détaillée dans le chapitre 5. On peut déjà signaler qu'elle est la méthode qui donne les meilleurs résultats à l'heure actuelle, si l'on considère le rapport qualité/temps de rendu. En effet, la méthode demande peu de contraintes du point de vue modélisation du milieu. On peut obtenir une précision suffisante sur la modélisation en raffinant la grille de voxel. On peut également réduire le coût mémoire de cette modélisation en utilisant une structure de type *Octree* pour décrire adaptativement le milieu. Aucune hypothèse n'est faite sur l'homogénéité et l'isotropie du milieu. Les temps de calcul restent cependant importants, et nos travaux ont porté sur leur minimisation tout en conservant la même qualité.

3.4 Conclusion

Nous avons présenté un tour d'horizon des méthodes d'illumination globale existantes en synthèse d'image classique. On s'aperçoit rapidement que les méthodes déterministes telles que la radiosité offrent de bons résultats en terme de précision. Cependant, elles sont mises en échec par la complexité des systèmes d'équations à résoudre. En effet, le nombre d'équations dépend exponentiellement du nombre de surfaces et/ou de volumes dans la scène. Les temps de calcul explosent donc pour des scènes complexes, rendant ces méthodes inapplicables dans la plupart des cas. De plus, malgré de nombreuses tentatives pour résoudre les problèmes de discontinuités des ombres dans ces méthodes, leur prise en compte est beaucoup plus difficile à mettre en œuvre qu'avec les méthodes basées sur le lancer de rayons, méthode qui a été proposée à l'origine spécialement pour résoudre les problèmes d'ombrage. L'introduction de processus stochastiques dans les méthodes de lancer de rayons (méthode de Monte Carlo) a grandement

participé à répandre ces méthodes. On a alors pu régler facilement le compromis entre précision et temps de calcul en jouant sur le nombre d'échantillons. Les temps de calcul restant relativement importants, plusieurs évolutions ont été proposées, les dernières et plus intéressantes étant les méthodes basées sur la propagation de particules à travers la scène et l'estimation de leur densité. La méthode la plus répandue à l'heure actuelle est sans aucun doute la méthode des cartes de photons proposée par Jensen [Jen01]. En effet, elle est très facile à mettre en œuvre et est autant capable de prendre en compte les surfaces que les milieux participants. Pour en revenir à notre problématique infrarouge, nous avons donc choisi d'adapter cette dernière au rendu spectral. Nous présentons dans le chapitre 4 les évolutions qui ont été nécessaires, et dans le chapitre 5 nous proposons une optimisation spécialement dédiée au rendu de milieux participants visant à minimiser le temps de rendu qui reste important avec la méthode de Jensen.

Chapitre 4

Lancer de photons multi-passes

4.1 Introduction

La méthode des cartes de photons était la méthode qui semblait la plus adaptée, en terme de compromis performances/qualité, au problème de l'illumination globale. Dans notre problématique du rendu optronique, nous avons choisi d'utiliser cette méthode, non seulement pour le rendu visible, mais aussi pour le rendu infrarouge. Nous proposons une évolution spectrale (section 4.3) qui permet d'uniformiser le rendu des deux domaines. La méthode originale étant limitée par la consommation mémoire pour le stockage des photons, son adaptation spectrale l'est d'autant plus, vu le nombre de bandes spectrales à stocker pour chaque photon. Après avoir décrit précisément la méthode originale des cartes de photons (section 4.2), nous décrivons les différents modèles spectraux que nous avons envisagé d'utiliser et proposons l'évolution spectrale de la méthode (section 4.3). Enfin, dans la section 4.4.3 nous présentons notre approche multi-passes qui permet de se détacher de la limitation mémoire intrinsèque à la méthode tout en conservant la même qualité.

4.2 Détails de la méthode des cartes de photons

Nous allons détailler ici le fonctionnement de la méthode des cartes de photons [Jen95]. Cela permettra de mettre en évidence les évolutions qui ont été nécessaires pour l'adaptation au rendu infrarouge. Nous commençons par expliquer ce que sont les photons, puis nous décrivons leur émission et leur propagation dans la scène qui permet de construire la carte de photons. Enfin nous expliquons la phase de reconstruction de la luminance à partir de la carte.

4.2.1 Description d'un photon

Avant de décrire le fonctionnement de la méthode, il est nécessaire de décrire ce que nous entendons par photon. Par analogie à la théorie corpusculaire de la lumière, un photon est une particule virtuelle qui est propagée le long des chemins lumineux à travers la scène. Informatiquement parlant, le photon est représenté par une structure de donnée qui contient :

- la position du photon. Elle correspond au point d'impact du photon sur les surfaces de la scène, i.e. au point d'intersection du rayon transportant le photon avec ces surfaces ;
- la direction incidente du photon. C'est-à-dire le vecteur directeur du rayon le portant ;
- l'énergie (ou puissance) du photon. Il s'agit d'un triplet RVB qui représente l'énergie pour chacune des couleurs.

La figure 4.1 présente la représentation conceptuelle d'un photon.

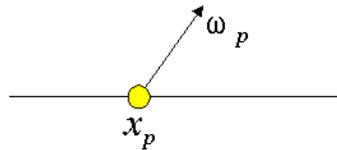


Figure 4.1. Représentation d'un photon. x_p est la position du photon dans l'espace 3D. ω_p est la direction incidente du photon.

4.2.2 Emission des photons

Les photons sont tirés aléatoirement depuis les sources lumineuses. Chaque source est caractérisée par une puissance d'émission, une géométrie et parfois par une fonction d'émission qui indique les positions et directions d'émission privilégiées. Pour des raisons de performances et de capacité de stockage, le nombre de photon qui peuvent être émis est limité. C'est pourquoi une stratégie de tirage par importance est utilisée. Celle-ci est basée sur une densité de probabilité qui met en rapport la puissance de chaque source avec la puissance totale. Un premier tirage est donc effectué pour choisir la source lumineuse en fonction de sa puissance. Ensuite un tirage est effectué pour choisir la position d'émission du photon sur la source. Ce tirage est effectué en échantillonnant par importance la fonction d'émission. Cependant, à l'heure actuelle, il existe encore peu de mesures de fonction d'émission. Dans le cas classique, on suppose que l'émission est uniforme. On tire donc uniformément un point sur la surface de la source. On tire ensuite la

direction du photon par rapport à une distribution en cosinus pour prendre en compte l'angle solide sous lequel est vu la source. En effet, un photon qui quitte la source orthogonalement à celle-ci, il porte plus d'énergie qu'un photon qui part en incidence rasante car une plus grande proportion de la source est visible depuis la surface éclairée. Le tirage en cosinus permet de donner plus d'importance aux directions qui éclairent le plus la scène. La figure 4.2 présente le mécanisme d'émission des photons.

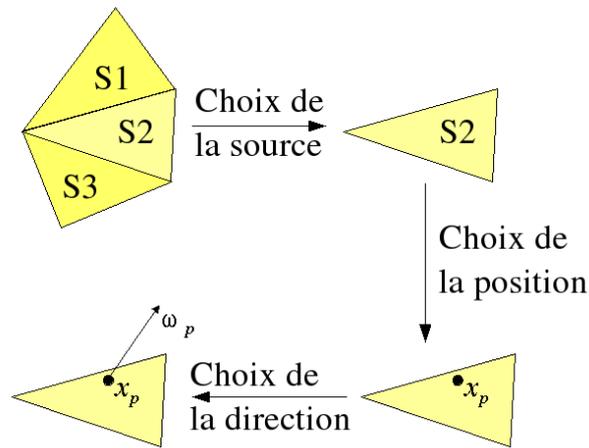


Figure 4.2. Principe de l'émission des photons.

4.2.3 Propagation des photons

Une fois la position x_p et la direction ω_p du photon p choisie, l'intersection I_p du rayon porteur $R_p = (x_p, \omega_p)$ avec la scène est calculée comme en lancer de rayon classique. La position du photon est remplacée par I_p . Le photon est alors stocké dans la carte de photons (cf section 4.2.4).

Ensuite, la méthode de la Roulette Russe est utilisée pour choisir, en fonction des caractéristiques de la surface S intersectée, si le photon est réfléchi ou absorbé. Le coefficient de réflectance étant l'intégrale de la FDRB sur l'espace des direction incidentes, cette méthode consiste à tirer uniformément un nombre aléatoire ξ entre 0 et 1, et à comparer ce nombre avec les coefficients de réflectances diffuse ρ_d et spéculaire ρ_s du matériau :

- si $\xi < \rho_d$ alors le photon subit une réflexion diffuse,
- sinon si $\xi < \rho_d + \rho_s$ alors le photon subit une réflexion spéculaire,
- sinon le photon est absorbé.

Si le photon est réfléchi de manière diffuse, on tire une nouvelle direction aléatoire ω_r uni-

formément distribuée sur l'hémisphère des directions sortantes, et la puissance du photon est multipliée par le coefficient de réflectance ρ_d . Si le photon est réfléchi spéculairement, alors la nouvelle direction est la direction de réflexion parfaite. Sa puissance est multipliée par ρ_s .

Une fois le photon mis à jour, il est à nouveau propagé dans la scène jusqu'à son absorption. Lorsqu'il est absorbé, un nouveau photon est émis depuis la source, et ainsi de suite jusqu'à avoir atteint le nombre de photons spécifié par l'utilisateur. Une optimisation proposée par Jensen est de ne pas stocker les photons primaires dans la carte. Nous verrons dans la section 4.2.5 que l'éclairage direct peut être calculé sans la carte de photons.

L'avantage de l'utilisation de la Roulette Russe est d'assurer que tous les chemins lumineux sont terminés sans introduire de biais supplémentaire dans la solution.

4.2.4 Stockage des photons

La carte de photons est utilisée lors de la reconstruction pour rechercher les k plus proches photons du point d'estimation. La structure de données la plus rapide pour ce type de recherche est sans conteste la grille régulière 3D. Néanmoins, il est nécessaire de stocker de nombreux photons pour obtenir une qualité satisfaisante. Or le stockage dans une grille régulière est très coûteux, car elle nécessite une discrétisation importante dans les zones où les photons sont nombreux. Jensen propose donc d'utiliser un *kdtree* équilibré [Ben75], qui propose le meilleur compromis entre coût de stockage et temps de recherche. Ces arbres se basent sur la division de l'espace. A chaque étage de l'arbre correspond une division selon une dimension (x, y, ou z en 3D). En général, on choisit de diviser l'espace selon la direction la plus longue. Pour obtenir un arbre équilibré, on coupe la dimension choisie de façon à avoir le même nombre de photons stockés de chaque côté de la division. L'équilibre de l'arbre garantit que le temps de recherche de k photons dans un arbre de N photons est de $O(k + \log N)$. Cependant ce temps est encore plus court lorsque les photons sont rapprochés dans l'espace. La figure 4.3 présente un exemple de subdivision pour un *kdtree* en deux dimensions.

4.2.5 Reconstruction de la luminance

Une fois, l'illumination précalculée et stockée dans la carte de photons, des rayons (O, ω_r) sont tracés depuis l'oeil à travers les pixels de l'écran. L'éclairage indirect en ce point x est

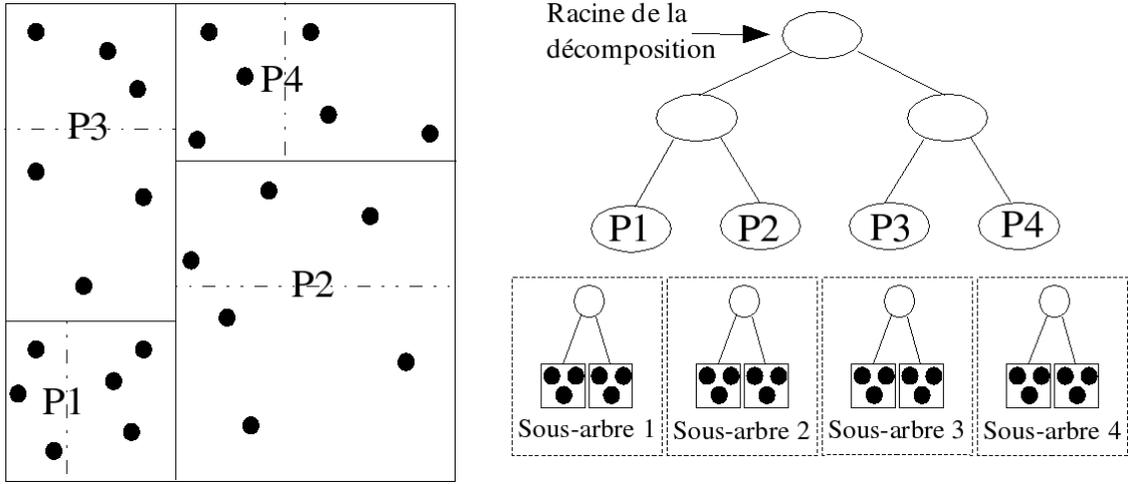


Figure 4.3. Exemple de kd-tree en deux dimensions

reconstruit en utilisant la méthode des k plus proches voisins :

$$L_{ind}(x, \omega_r) = \sum_{p=1}^k \frac{f_r(x, \omega_p, \omega_r) \Delta \phi_p}{\Delta A} \quad (4.1)$$

où $\Delta \phi_p$ est la puissance du photon p , ΔA l'aire associée à l'élément de surface S centré en x et f_r la FDRB. Ici, seule la luminance indirecte est calculée en utilisant la carte de photons. Bien qu'il soit possible de calculer l'intégralité de l'éclairage de cette manière, le nombre de photons à utiliser pour réduire le bruit basse fréquence est très grand. Jensen préfère donc utiliser une optimisation pour calculer l'éclairage direct. Etant donné que dans la plupart des scènes visibles, les sources lumineuses sont clairement identifiées, Jensen évalue la luminance directe en envoyant un rayon *d'ombre* vers la source lumineuse. On a alors :

$$L_{dir}(x, \omega_r) = \sum_{l=1}^{N_l} f_r(x, \omega_l, \omega_r) L_l(x_l, \omega_l) \cos \theta_l \Delta \omega_l \quad (4.2)$$

où N_l est le nombre de sources, $L_l(x_l, \omega_l)$ est la luminance émise par le point x_l choisi sur la source dans la direction $\omega_l = \frac{x-x_l}{\|x-x_l\|}$ associée à l'angle solide élémentaire $\Delta \omega_l$. Cependant, nous verrons dans la section 4.3.5 que cette optimisation est limitative sur le nombre de sources qui peuvent être traitées.

Finalement, la luminance totale est :

$$L(x, \omega_r) = \sum_{l=1}^{N_l} f_r(x, \omega_l, \omega_r) L_l(x_l, \omega_l) \cos \theta_l \Delta \omega_l + \sum_{p=1}^k \frac{f_r(x, \omega_p, \omega_r) \Delta \phi_p}{\Delta A} \quad (4.3)$$

Cette équation est celle de la visualisation directe de la carte de photons (figure 4.4). La solution peut néanmoins présenter un bruit basse fréquence si le nombre de photons n'est pas assez élevé.

Afin de réduire le nombre de photons nécessaires, Jensen propose de n'évaluer la luminance indirecte qu'au deuxième rebond des rayons de visualisation dans la scène. Nous ne nous étendrons pas sur cette optimisation, mais la figure 4.5 présente cette idée et pour plus d'information le lecteur peut se référer à [Jen01].

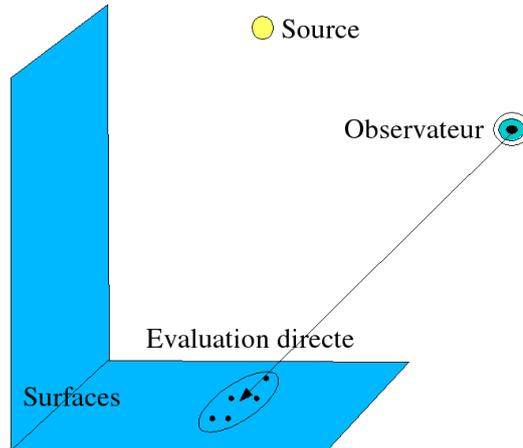


Figure 4.4. *Visualisation directe de la carte de photons*

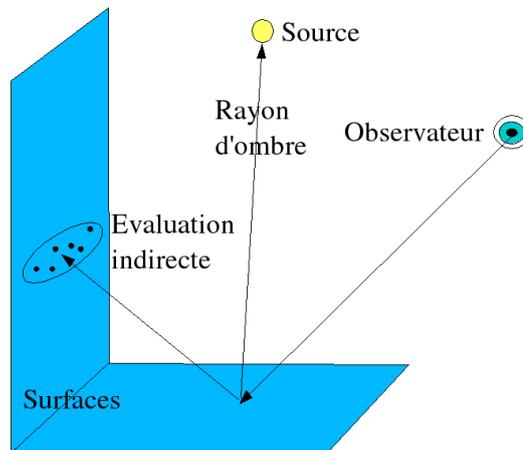


Figure 4.5. *Visualisation de la carte de photons au deuxième rebond*

4.3 Lancer de photons spectral

La méthode des cartes de photons a fait ses preuves dans la simulation du visible, et se pose comme l'une des méthodes les plus performantes pour ce type de rendu. Dans notre problématique du rendu infrarouge, il semblait donc logique d'adapter cette méthode au rendu spectral. Nous commençons donc par faire un inventaire des différentes représentations du

spectre lumineux existantes. Puis en justifiant notre choix de modèle spectral, nous décrivons les évolutions apportées en conséquence à la méthode. Nous préférons revenir à la terminologie de *lancer de photons* car la méthode des cartes de photons en constitue simplement une optimisation intéressante.

4.3.1 Modèle spectral

La majorité des méthodes de rendu que l'on trouve en synthèse d'image concernent uniquement la simulation visible. Ces méthodes ont été développées en prenant en compte le système visuel humain et la perception humaine (par exemple [CIP00], [RP97] et Devlin2002). Dans le cas de la simulation infrarouge, ces hypothèses n'ont pas de sens, vu que les mesures sont effectuées par des capteurs qui possèdent tous leurs propres caractéristiques, notamment au niveau spectral. La méthode du lancer de photons ne fait pas exception à la règle. Il est donc nécessaire de modifier la méthode pour qu'elle puisse effectuer des simulations spectrales physiques.

De nombreuses méthodes génériques ont été développées pour décrire des signaux naturels (cf. [Kay88, PW93] pour les idées générales) et sont applicables aux fonctions spectrales.

Echantillonnage par points Cette méthode représente une fonction définie sur une région continue par ses valeurs en un ensemble de points échantillonnés dans la région. Pour les fonctions spectrales, les points d'échantillonnage sont des longueurs d'ondes discrètes. Même s'il arrive que l'on utilise un échantillonnage non uniforme pour optimiser la représentation en respectant la perception humaine [Mey88, BB88, Zeg97], habituellement les points d'échantillonnage sont choisis uniformément dans toutes les mesures expérimentales [NNJ43, Kri53, Col79, VGI94] et les calculs [CT82, HG83, Hal89], avec un intervalle d'échantillonnage typiquement de 1 à 10 nm. Un avantage de cette méthode est sa capacité à atteindre une très grande précision si le nombre d'échantillons est suffisamment élevé (ou si l'intervalle d'échantillonnage est suffisamment petit). La méthode est aussi performante en temps de calcul. Pour le calcul du produit de deux fonctions spectrales, il suffit de multiplier entre eux les valeurs des fonctions pour les échantillons de mêmes longueurs d'onde. Pour N échantillons, le coût de calcul est $O(N)$, ce qui est très intéressant. De plus, cette méthode est portable car elle ne fait aucune hypothèse sur la forme du spectre. L'inconvénient de cette méthode est le coût mémoire. Plus on veut être précis, plus on utilise d'échantillons, et donc plus on a besoin d'espace mémoire pour stocker les échantillons. Si on n'utilise pas assez d'échantillons, l'erreur significative augmente. Un échantillonnage adaptatif

[DMCP94, RP97] a été proposé pour réduire le coût mémoire. Cependant, un grand nombre de points reste nécessaire pour les spectres présentant des pics pour certaines longueurs d'onde. De plus, l'échantillonnage adaptatif augmente la complexité du problème. L'échantillonnage non uniforme [Mey88] réduit, quant à lui, les erreurs sur des spectres lisses, mais n'élimine pas les problèmes dus à des spectres présentant des pics. Cette méthode se rapproche de la méthode raie par raie utilisée dans le monde de la physique.

Représentation polynomiale Les polynômes ont été considérés pour la représentation spectrale il y a très longtemps [Moo45]. L'idée a été reprise récemment pour l'image de synthèse [RF91]. Dans cette approche, une fonction spectrale est représentée par une collection de coefficients polynomiaux. La compacité de cette représentation dépend du degré polynomial utilisé. Cependant, quand le degré polynomial est trop élevé (typiquement supérieur ou égal à 7), une instabilité numérique apparaît due aux termes de puissance élevée [For57]. Les polynômes utilisés doivent donc être de degrés bas, ce qui limite la précision pouvant être atteinte. De plus, cette méthode n'est pas adaptée à la représentation de spectre présentant des pics. La complexité de la multiplication est en $O(M^2)$, avec M le degré maximum du polynôme utilisé. Enfin, même si la représentation polynomiale est portable, elle n'est pas flexible. Si on accroît le degré pour une meilleure précision, tous les coefficients doivent être recalculés.

Modèle linéaire Ce modèle a reçu beaucoup d'attention [SWO77, Hor84, BG84, MW86, Yui87, For90, DF92, D'Z92, MW92, VT92, FDF94, RGBHA97, GBNHAR98]. Le but est d'exprimer une fonction spectrale comme une combinaison linéaire d'un ensemble de fonctions de base linéairement indépendantes les unes des autres. En principe, n'importe quelles fonctions de base linéairement indépendantes peuvent être utilisées. Par exemple, les séries de Fourier ou les polynômes de Chebychev [IK66, GHJ96]. Dans la plupart des études, les fonctions de base sont dérivées numériquement de telle façon qu'elle représentent tout le spectre dans un domaine spécifique [Mal86, DF92, Pee93]. Plus simplement, les fonctions de bases sont choisies en fonction des spectres des sources et des FDRB spectrales d'une scène. Ce modèle a des avantages de précision et de compacité. Le nombre de fonctions de base détermine la compacité. Il est aussi flexible car on peut choisir un nombre variable de fonctions de base pour représenter le spectre. Cependant, cette méthode n'est pas efficace. Tout d'abord, la multiplication spectrale est en $O(M^2)$, avec M le nombre de fonctions de base.

Modèle composite Sun [SFD98, SDF99] propose un modèle hybride pour la représentation du spectre lumineux. L'idée est de décomposer le spectre en deux parties qui sont représentées différemment. La partie lisse peut être exprimée comme une combinaison linéaire de fonctions de base, et ainsi être représentée par un ensemble de coefficients. Les pics sont représentés par des fonctions de type Dirac définies par leurs localisations et leurs hauteurs. Les fonctions de base sont analytiques et définies par quelques paramètres seulement sur un large spectre. Ce qui permet un stockage efficace des données spectrales. Une stratégie de ré-échantillonnage des valeurs de ces fonctions est recommandée afin d'optimiser les performances, notamment pour la multiplication spectrale. Le nombre de points de ré-échantillonnage peut être déterminé en fonction de l'importance relative de la précision, de la compacité et de la performance désirées.

Modèle des k -distributions Ce modèle spectral n'est pas encore utilisé dans le monde de la synthèse d'image. Par contre, il est le modèle le plus utilisé dans le monde de la physique en ce qui concerne le transfert radiatif. Sans nous étendre sur le sujet, nous souhaitons donner un bref aperçu de ce modèle qui semble pouvoir devenir à terme le modèle le plus pratique pour la simulation informatique des milieux participants.

La méthode des k -distributions a été développée pour calculer la transmission spectrale, et par conséquent l'intensité spectrale ou les flux spectraux. Elle se base sur un groupement des coefficients d'absorption des milieux participants. Lors du calcul de la transmission, une intégration selon la longueur d'onde est nécessaire. L'idée est de remplacer cette intégration par une intégration sur le coefficient d'absorption (noté k en physique). Pour une bande donnée, cela consiste à introduire une fonction $f(k)$ de distribution normalisée de k . $f(k)$ est la fraction de bande spectrale qui correspond aux valeurs d'absorption comprise entre k et $k + dk$. La figure 4.6 illustre la réorganisation nécessaire. Finalement, les résultats obtenus au niveau de la bande de longueur d'ondes sont les mêmes que si on avait effectué une intégration raie par raie. L'utilisation de ce modèle a été proposé par Domoto en 1974 [Dom74] et [Fom04] en propose une version améliorée.

4.3.2 Choix du modèle spectral

Au vu de l'inventaire des modèles existants, le choix n'est pas immédiat, chacun possédant ses avantages et ses inconvénients. Cependant, on peut noter que, dans le monde de la physique, le modèle des k -distributions semble être celui qui a le vent en poupe. Néanmoins, le but de cette

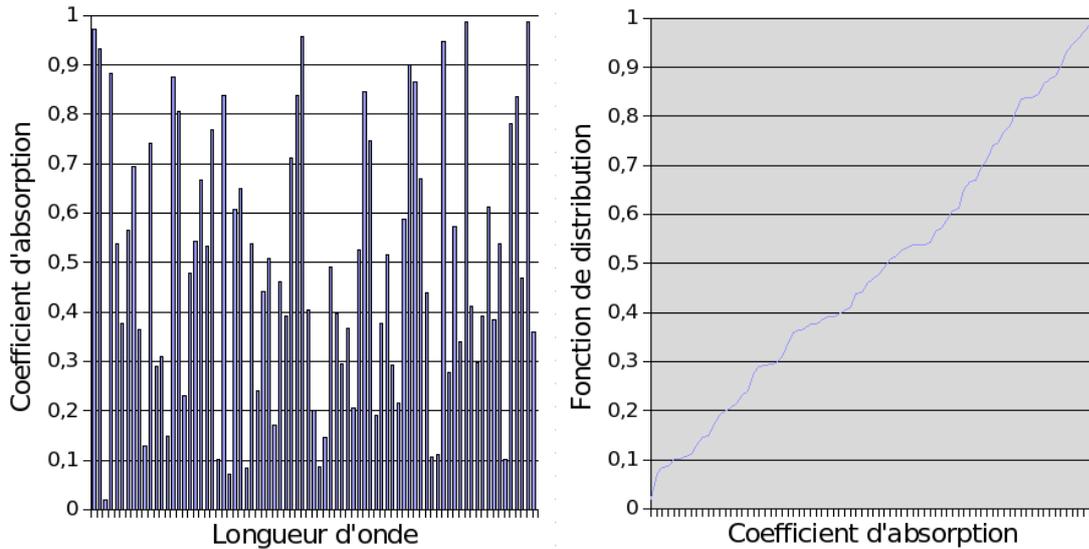


Figure 4.6. Réorganisation des coefficients d'absorption par la méthode des *k*-distributions.

thèse n'est pas de définir un modèle spectral mais de développer un algorithme de rendu général applicable à une majorité de modèles. La seule contrainte est qu'il existe une fonction de composition des spectres qui permette d'additionner et de multiplier les quantités spectrales entre elles. Cette hypothèse est vérifiée par la majorité des modèles présentés ci-dessus. Nous avons donc choisi d'utiliser le modèle le plus simple pour faciliter la compréhension de l'algorithme. Il s'agit d'une représentation du spectre par bandes grises (figure 4.7). Les lois de composition utilisées sont alors de simples additions et multiplications bande par bande.

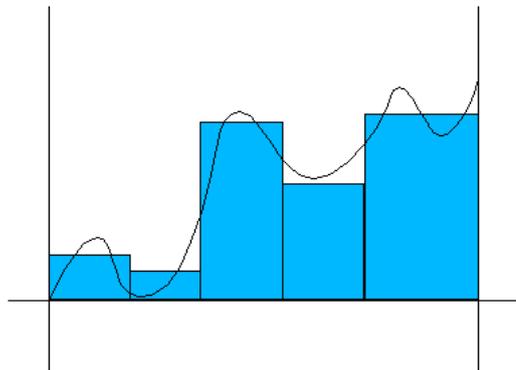


Figure 4.7. Approximation du spectre par un modèle de bandes grises

4.3.3 Photons spectraux

Dans [Jen01], un photon est une structure de donnée contenant une position, une direction et une puissance RVB. Similairement, nous utilisons des photons qui contiennent une position

une direction et un flux spectral représenté par un tableau de valeurs. Chaque case du tableau correspond à la puissance de chacune des bandes utilisées pour décomposer le spectre. Nous aurions pu utiliser un photon par bande, mais dans certains cas, nous pourrions avoir besoin de nombreuses bandes pour représenter le spectre lumineux avec suffisamment de précision. Il est donc intéressant de regrouper toutes les bandes dans un seul photon et ainsi, d'une part, factoriser le stockage des positions et des directions et d'autre part diminuer le nombre de chemins lumineux à évaluer. Cela offre également l'avantage non négligeable de diminuer le nombre de photons contenu dans la carte de photons, et donc de réduire le temps de recherche dans celle-ci. En effet, les temps de tri et de recherche augmentent avec la taille de la carte. De plus, ce choix est en accord avec la généralisation de l'algorithme pour différents modèles de spectre (section 4.3.2).

4.3.4 Réflexions spectrales

Les matériaux sont définis par leurs propriétés d'absorption, de diffusion. Toutes sont à caractère spectral. Nous choisissons de décrire ces propriétés en utilisant les mêmes bandes que celles utilisées pour le capteur et les photons. La passe de tracé de particules utilise la méthode de la Roulette Russe comme dans [Jen01], que nous avons adapté à notre modèle spectral. Nous utilisons un coefficient moyenné au lieu du coefficient unique utilisé pour le rendu mono-bande. Pour un matériau donné, nous calculons les coefficients moyennés de réflectance diffuse ($\overline{\rho_d}$) et spéculaire ($\overline{\rho_s}$) :

$$\overline{\rho_d} = \frac{1}{\Delta\Lambda} \int_{\Delta\Lambda} \rho_{d_\lambda} d\lambda$$

$$\overline{\rho_s} = \frac{1}{\Delta\Lambda} \int_{\Delta\Lambda} \rho_{s_\lambda} d\lambda$$

où $\Delta\Lambda$ est la somme des largeurs de bandes. Ces coefficients sont utilisés par la Roulette Russe pour déterminer si le photon est absorbé ou réfléchi. Etant donné un nombre ξ tiré aléatoirement entre 0 et 1 :

- si $\xi \in [0, \overline{\rho_d}]$ alors le photon subit une réflexion diffuse,
- si $\xi \in]\overline{\rho_d}, \overline{\rho_d} + \overline{\rho_s}]$ alors le photon est réfléchi spéculairement,
- si $\xi \in]\overline{\rho_d} + \overline{\rho_s}, 1]$ alors le photon est absorbé.

Pour prendre en compte le fait que le type de réflexion devrait avoir été choisi selon la réflectance spectrale moyennée et assurer la conservation de l'énergie, nous devons mettre à l'échelle la

puissance spectrale de la particule réfléchiée en fonction du cas effectivement choisi. En fait, le choix du type de réflexion utilise une fonction de densité de probabilité (FDP [Dut01]) définie pour chaque bande $\Delta\lambda$ par :

- pour une réflexion diffuse : $FDP_d(\Delta\lambda) = \bar{\rho}_d$.
- pour une réflexion spéculaire : $FDP_s(\Delta\lambda) = \bar{\rho}_s$.

Selon le principe de l'échantillonnage par importance, les poids des photons doivent alors être modifiés pour prendre en compte l'échantillonnage par importance utilisant la FDP précédente. Pour une réflexion diffuse et pour chaque bande $\Delta\lambda$, on a :

$$\Phi_{r,\Delta\lambda} = \Phi_{i,\Delta\lambda} \frac{\rho_{d,\Delta\lambda}}{FDP_d(\Delta\lambda)} = \Phi_{i,\Delta\lambda} \frac{\rho_{d,\Delta\lambda}}{\bar{\rho}_d}$$

où $\Phi_{i,\Delta\lambda}$ est la puissance du photon incident pour la bande $\Delta\lambda$ et $\Phi_{r,\Delta\lambda}$ est la puissance du photon réfléchi. Afin de l'adapter aux autres modèles spectraux, il suffit de redéfinir le calcul des coefficients de réflectances moyennés ainsi que la fonction de densité de probabilité pour ces modèles. Aucune modification de l'algorithme n'est nécessaire.

4.3.5 Estimation spectrale de la luminance

Notre méthode de rendu diffère un peu des méthodes classiques d'estimation de densité. Des rayons primaires sont tracés dans la scène depuis l'origine du capteur à travers la grille de pixels. La luminance spectrale est calculée à leur intersection avec la scène. Dans notre problématique, nous ne pouvons pas utiliser l'optimisation portant sur le calcul de l'illumination directe qui consiste à tracer un ou plusieurs rayons vers chaque source lumineuse. Or, comme nous l'avons vu dans le chapitre 1, la loi du corps noir est très sensible à la température dans le domaine infrarouge. Par conséquent, toutes les surfaces de la scènes sont source en infrarouge car elles sont émettrices pour la température ambiante. Le nombre de sources est donc nécessairement très grand. Il est donc impossible, sans faire s'effondrer les performances, d'échantillonner toutes les sources lumineuses (un rayon devrait être lancé vers chaque surface de la scène pour chaque point d'estimation). Nous calculons donc l'intégralité de l'illumination globale par estimation de densité dans la carte de photons.

Habituellement, la méthode des cartes de photons est utilisée uniquement pour le calcul de l'éclairage indirect. En effet, celui-ci varie lentement (basses fréquences) et la méthode est particulièrement adaptée à ce type de fréquence (estimation de densité par noyau). Par contre, l'éclairage direct varie en haute fréquence. Il est donc nécessaire d'augmenter le nombre de

photons afin de l'échantillonner correctement. En effet, le théorème de Shannon indique que la fréquence minimale d'échantillonnage (i.e. le nombre de photons) d'un signal (i.e. éclairage direct) doit être au moins le double de sa fréquence maximale. La mémoire nécessaire est donc dans notre cas plus importante que dans la méthode de Jensen.

Les luminances de toutes les bandes sont calculées simultanément en utilisant le calcul vectoriel. Pour une bande donnée $\Delta\lambda$, nous avons adapté l'équation du rendu :

$$L_{r,\Delta\lambda}(x, \vec{\omega}_r) = \int_{\Omega} f_{r,\Delta\lambda}(\vec{\omega}_i, \vec{\omega}_r) L_{i,\Delta\lambda}(x, \vec{\omega}_i) (\vec{n}_x \cdot \vec{\omega}_i) d\omega_i$$

où $L_{r,\Delta\lambda}$ est la luminance réfléchie dans la direction $\vec{\omega}_r$ pour la bande $\Delta\lambda$. Ω est l'hémisphère des directions incidentes au point x , $f_{r,\Delta\lambda}$ est la FDRB au point x pour la bande $\Delta\lambda$ et $L_{i,\Delta\lambda}$ la luminance incidente pour la bande $\Delta\lambda$. La carte de photons contient les informations de flux. En utilisant la relation entre le flux et la luminance :

$$L_{i,\Delta\lambda}(x, \vec{\omega}_i) = \frac{d^2\Phi_{i,\Delta\lambda}(x, \vec{\omega}_i)}{(\vec{n}_x \cdot \vec{\omega}_i) d\omega_i dA_i}$$

l'équation du rendu devient :

$$L_{r,\Delta\lambda}(x, \vec{\omega}_r) = \int_{\Omega} f_{r,\Delta\lambda}(\vec{\omega}_i, \vec{\omega}_r) \frac{d^2\Phi_{i,\Delta\lambda}(x, \vec{\omega}_i)}{dA_i}$$

Le flux incident $\Phi_{i,\Delta\lambda}$ est estimé en localisant dans la carte de photons les n plus proches photons du point x . On a alors :

$$L_{r,\Delta\lambda}(x, \vec{\omega}_r) \approx \sum_{p=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}_r) \frac{\Delta\Phi_{p,\Delta\lambda}(x, \vec{\omega}_p)}{\Delta A}$$

où $\Delta\Phi_{p,\Delta\lambda}$ est la puissance du photon p pour la bande $\Delta\lambda$, ΔA est l'aire de la surface où se situent les photons utilisés pour l'estimation du flux (i.e. disque support de l'estimation de densité) et $\vec{\omega}_p$ est la direction du photon.

D'une manière générale on a :

$$L_r(x, \vec{\omega}_r) \approx \sum_{p=1}^n f_r(x, \vec{\omega}_p, \vec{\omega}_r) \frac{\Delta\Phi_p(x, \vec{\omega}_p)}{\Delta A}$$

puisque les propriétés des modèles spectraux le permettent. Ici encore il suffit d'avoir implémenté l'addition et la multiplication sur les spectres pour changer de modèle.

4.4 Approche multi-passes

4.4.1 Problèmes dus à l'estimation de densité spectrale appliquée au rendu infrarouge

4.4.1.1 Bruit spéculaire

A la suite de tests, nous avons noté quelques problèmes inhérents à notre méthode spectrale. La figure 4.8 présente le rendu infrarouge d'un tube possédant un matériau émissif et spéculaire. Le rendu a été effectué avec un million de photons. Le tube est l'unique source de la scène.

Le bruit dans l'image générée est important car la densité de photons n'est pas suffisante pour estimer la luminance spéculaire du matériau (cf figure 4.9). Lors de l'estimation de la luminance, la puissance des photons est modulée par la valeur de la FDRB. Comme la densité de photon est insuffisante, la probabilité de trouver des photons, dont la direction d'incidence est dans le cône spéculaire de la FDRB, est faible. Elle dépend de la largeur du cône spéculaire et de la densité de photons. La solution à ce problème est alors d'augmenter la densité de photons et par conséquent de tracer et stocker un plus grand nombre de photons, ce qui augmente la quantité de mémoire nécessaire.

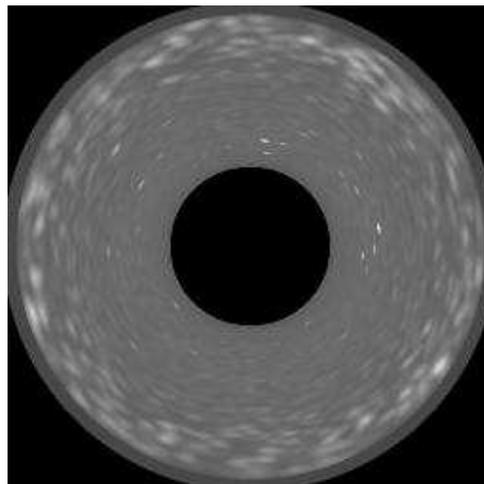


Figure 4.8. Rendu infrarouge d'un tube émissif et spéculaire, vu dans l'axe, en utilisant un million de photons.

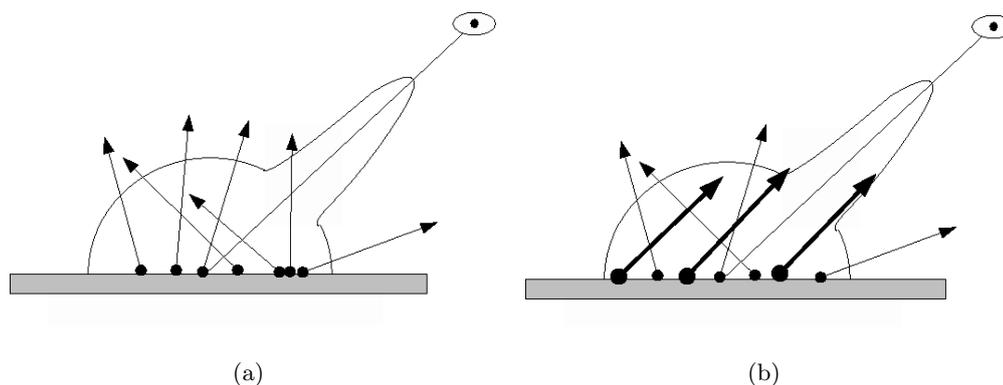


Figure 4.9. Problème d'estimation de densité pour les matériaux spéculaires. (a) : Aucun photon n'a sa direction dans le cône spéculaire. (b) : Les photons en gras ont leur direction dans le cône spéculaire.

4.4.1.2 Consommation mémoire

Cependant, stocker un plus grand nombre de photons a un coût non négligeable. Mesurons la taille d'un photon spectral. Tout d'abord, la position et la direction sont stockés dans 6 réels avec double précision (*double*). Ensuite, la puissance de chaque bande est stockée dans 1 *double*. Donc, si on considère une bande unique, la taille du photon est de 7 *double* ou 56 octets. Cela signifie que l'on a besoin de 56 Méga-octets pour stocker un million de photons. Mais ceci est donné pour une seule bande. Si le nombre de bandes est 100, alors nous avons besoin de 848 octets par photon, et donc 848 Mo pour un million de photons. Ceci est prohibitif pour la majorité des ordinateurs. Nous devons donc trouver une solution pour augmenter le nombre de photons pour l'estimation de densité, sans pour autant augmenter la consommation mémoire. Bien entendu, si nous avons choisi d'utiliser des réels simple précision (*float*) à la place des *double*, ou si nous avons choisi d'utiliser une méthode de compression pour stocker le flux spectral (similaire au format RGB à exposant partagé de Ward [Jen01]) ou encore un modèle spectral possédant une représentation avec un coût de stockage plus faible, nous diminuerions la consommation mémoire. Mais, d'une part, nous voulons assurer la précision sur les calculs, aussi bien pour la construction des chemins lumineux que pour le calcul de la luminance, d'autre part, nous pensons que le problème reste le même, vu que nous devons stocker 20 millions de photons ou plus dans la plupart des cas pour obtenir une qualité suffisante.

Notre objectif est finalement de trouver un moyen d'augmenter le nombre de photons stockés dans la carte. Mais si le nombre de particules et/ou le nombre de bandes est trop grand, nous

n'avons pas assez de mémoire pour stocker la carte de photon entière. Nous devons donc mettre une partie de la carte de photons en cache sur le disque. Nous voulons également préserver les performances de la méthode. Notre objectif consiste donc à optimiser la gestion de cache de la carte de photons.

4.4.2 Méthodes de cache existantes

Ward a proposé la méthode de cache d'éclairement (*Irradiance Caching* [WRC88]) en 1988. Dans sa méthode, il n'utilise l'estimation de densité que pour calculer l'illumination indirecte. Son idée est de tirer avantage du fait que l'illumination indirecte change doucement. Il précalcule donc l'éclairement pour un certain nombre de points de la scène. Le nombre de points utilisés dépend de la régularité de l'illumination indirecte. En effet, il utilise une estimation du gradient de luminance pour décider si la densité de points d'estimation est suffisante ou s'il faut en ajouter. Ensuite, l'éclairement est interpolé lors du rendu. Cette méthode est très efficace pour une majorité de scènes et pourrait certainement être adaptée au rendu infrarouge. Comme nous l'avons vu dans la problématique, il existe trois découpages classiques du spectre lumineux pour le rendu en infrarouge : bande I ($[1.10^{-6}, 2.10^{-6}]$ mètres), bande II ($[3.10^{-6}, 5.10^{-6}]$ mètres) et bande III ($[8.10^{-6}, 12.10^{-6}]$ mètres). Pour notre méthode, le cache d'éclairement serait adapté pour la simulation en bandes II et III car l'illumination directe varie lentement, toutes les surfaces de la scène étant émissives. Malheureusement, en bande I, l'illumination directe peut être aussi irrégulière que dans le spectre visible. Pour ce type de rendu, nous ne pouvons pas utiliser cette méthode car nous calculons la totalité de l'illumination par évaluation directe de la carte de photons et que l'illumination directe peut varier rapidement. L'hypothèse de variation lente de l'éclairement n'est alors pas vérifiée dans le cas qui nous intéresse. En effet, le nombre de points d'estimation que nous devrions utiliser pourrait être très grand pour certaines scènes.

Dans les méthodes [SWH⁺95] et [Wal98] vues dans le chapitre 3, Shirley et Walter ont proposé une méthode indépendante du point de vue en pré-calculant l'éclairage sur un maillage. Une phase de tracé de particules, puis une phase d'estimation de densité sont utilisées pour calculer l'illumination sur le maillage de la scène. Ensuite, une phase de triangulation est utilisée pour produire un maillage d'illumination où celle-ci peut-être considérée comme linéaire sur chaque face du maillage. Ensuite, le maillage utilisé peut-être affiché en utilisant un affichage standard (OpenGL par exemple, ou un lancer de rayons). Comme le maillage d'illumination dépend de la complexité de la scène et de la régularité de l'illumination, cette méthode est dépendante des

deux.

Pour ces deux méthodes, il existe des cas où elles sont inapplicables en relation avec la complexité de la scène et de la régularité de l'éclairage. Par exemple, une scène possédant un grand nombre d'objets petits et complexes (typiquement un moteur) peut présenter une illumination très irrégulière qui oblige à augmenter dangereusement les tailles des structures utilisées pour précalculer l'illumination dans ces méthodes.

Or, nous cherchons une méthode capable de rendre des scènes complexes possédant une illumination potentiellement irrégulière sans augmenter la complexité de l'algorithme. Nous devons donc trouver une autre solution.

4.4.3 Principe de base de l'approche multi-passes.

La méthode que nous proposons possède un principe simple. Comme nous ne pouvons pas stocker suffisamment de photons en mémoire pour atteindre une précision suffisante, nous avons pensé à effectuer plusieurs estimations de densité qui utilisent des cartes de photons tenant en mémoire. Ces passes d'estimation peuvent être effectuées les unes après les autres. Néanmoins, cela ne fonctionne que dans le cas où les petites cartes de photons utilisées forment une partition de la grosse carte de photons complète que nous aurions dû utiliser. Pour assurer cela, nous utilisons une graine différente pour chaque passe de construction de la carte de photons afin d'initialiser les chaînes de tirages pseudo-aléatoires. En effet, simplement modifier la graine des tirages assure dans la majorité des cas que l'ensemble des chemins calculés sont différents. Les probabilités de recouvrement de cartes de photons existent mais elles sont très réduites et celles-ci n'introduisent que très peu de biais dans la solution.

Nous utilisons un cache pour stocker les luminances calculées à chaque passe. Ce cache est un tableau de luminances spectrales de mêmes dimensions que l'image finale. Chaque case du tableau correspond à un point d'estimation (i.e. un pixel de l'image capteur). Les points d'estimation sont les mêmes pour chaque passe car nous calculons la même image de nombreuses fois avec des cartes de photons différentes. Donc la taille du cache dépend uniquement de la taille de l'image (pour un découpage spectral donné). Par exemple, une image de 1024×768 pixels calculée en utilisant 100 bandes nécessite un cache de 630 Mo, ce qui est inférieur à la taille de la carte de photons nécessaire (qui fait 848 Mo comme vu dans la section 4.4.1.2). De plus, nous pouvons, dans ce cache unique, stocker l'illumination correspondant à autant de photons que nécessaires. Finalement, les temps de lecture et de sauvegarde totaux du cache sur le disque dur

sont plus rapides. En effet, si l'on enregistre la carte globale dans un cache, de nombreux défauts de cache peuvent arriver lors de la recherche des photons pour chaque point d'estimation. La carte est chargée par zone de l'espace et les défauts de cache sont importants pour les points d'estimation situés à la limite de ces zones. Dans notre méthode, chaque carte étant globale, la lecture et l'écriture du cache n'est nécessaire qu'à la fin de chaque passe. De plus, Larsen et Christensen [LC03] ont montré, pour un but différent du notre, qu'utiliser plusieurs cartes de photons de petites taille est plus rapide que d'utiliser une grosse carte unique.

La figure 4.10 montre un exemple. En fait, une estimation de densité de la luminance en un point d'échantillonnage en cherchant 600 photons dans une carte de deux millions de photons est égale à la moyenne de deux estimations de densité cherchant 300 photons dans deux cartes de photons distinctes d'un million de photons. La démonstration de ceci est exposée dans la prochaine section. La figure 4.11 présente l'équivalence des résultats obtenus entre une et trente passes.

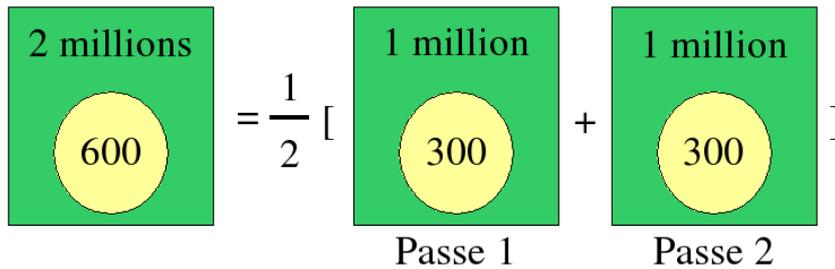


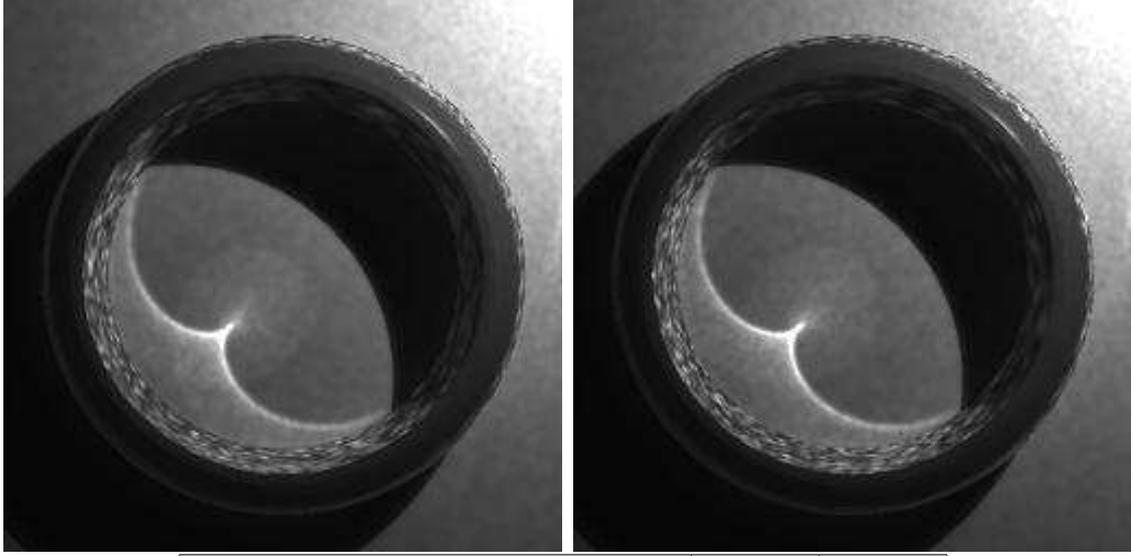
Figure 4.10. Principe du lancer de photons multi-passes. Il est équivalent de calculer la luminance en utilisant une estimation avec 600 photons cherchés dans une carte de 2 millions, et de calculer la luminance comme la moyenne de deux estimations de 300 photons cherchés dans deux cartes de 1 million, les deux cartes formant une partition de la grande.

4.4.4 Reconstruction de l'image finale

Un pixel de l'image définit un rayon unique partant de l'oeil, et donc définit un point d'intersection unique avec la scène. Nous avons vu dans la section 4.3.5 que la luminance en ce point est donnée par :

$$L_{r,\Delta\lambda}(x, \vec{\omega}) \approx \sum_{p=1}^{n_r} f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}) \frac{\Delta\Phi_{p,\Delta\lambda}(x, \vec{\omega}_p)}{\Delta A}$$

où n_r est le nombre de photons utilisés pour l'estimation de densité.



	Gauche	Droite
Nombre de passes	1	30
Taille de la carte de photons	3M	100K
Nombre de photons recherchés	600	20
Luminance moyenne des pixels	1.39736	1.39929
Ecart-type de la luminance des pixels	1.19078	1.19402

Figure 4.11. Illustration de l'équivalence entre un rendu avec une (gauche) et trente passes (droite).

Prenons l'hypothèse que l'énergie lumineuse, $\Phi_{\Delta\lambda}$, de la bande $\Delta\lambda$ est également distribuée entre tous les photons :

$$\Phi_{p,\Delta\lambda}(x, \vec{\omega}_p) = \frac{\Phi_{\Delta\lambda}}{N_t}$$

où N_t est le nombre total de photons.

On a alors :

$$L_{r,\Delta\lambda}(x, \vec{\omega}) \approx \sum_{p=1}^{n_r} f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{N_t \Delta A}$$

On suppose maintenant qu'il existe un entier m tel que $N_t = mN$ et $n_r = mn$. Alors on a :

$$L_{r,\Delta\lambda}(x, \vec{\omega}) \approx \sum_{p=1}^{mn} f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A}$$

Nous pouvons maintenant diviser la somme en m termes :

$$\begin{aligned}
L_{r,\Delta\lambda}(x, \vec{\omega}) \approx & \left[\sum_{p=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A} \right. \\
& + \sum_{p=n}^{2n} f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A} \\
& + \dots \\
& \left. + \sum_{p=n(m-1)}^{nm} f_{r,\Delta\lambda}(x, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A} \right]
\end{aligned}$$

On fait un changement de variables pour chaque somme, et on suppose que pour chaque somme, les photons $\vec{\omega}_{p_i}$ sont dans la carte de photons i et les cartes de photons sont distinctes. Alors nous avons :

$$\begin{aligned}
L_{r,\Delta\lambda}(x, \vec{\omega}) \approx & \left[\sum_{p_1=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_{p_1}, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A} \right. \\
& + \sum_{p_2=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_{p_2}, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A} \\
& + \dots \\
& \left. + \sum_{p_m=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_{p_m}, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A} \right]
\end{aligned}$$

On peut reformuler cela en une somme :

$$L_{r,\Delta\lambda}(x, \vec{\omega}) \approx \sum_{i=1}^m \sum_{p_i=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_{p_i}, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{Nm\Delta A}$$

On factorise :

$$L_{r,\Delta\lambda}(x, \vec{\omega}) \approx \frac{1}{m} \sum_{i=1}^m \sum_{p_i=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_{p_i}, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{N\Delta A} \quad (4.4)$$

Le terme $\sum_{p_i=1}^n f_{r,\Delta\lambda}(x, \vec{\omega}_{p_i}, \vec{\omega}) \frac{\Phi_{\Delta\lambda}}{N\Delta A}$ correspond à une estimation de densité cherchant n photons d'une carte de N photons.

Comme dans la section 4.3.5, nous pouvons écrire l'équation plus générale permettant de s'adapter à aux différents modèles spectraux :

$$L_r(x, \vec{\omega}) \approx \frac{1}{m} \sum_{i=1}^m \sum_{p_i=1}^n f_r(x, \vec{\omega}_{p_i}, \vec{\omega}) \frac{\Phi}{N\Delta A} \quad (4.5)$$

L'équation montre que calculer la luminance avec m petites cartes contenant N photons en utilisant n photons pour l'estimation de densité puis de calculer une moyenne équivaut à calculer la luminance avec un grosse carte de $N_t = \frac{N}{m}$ en utilisant $n_r = \frac{n}{m}$ photons pour l'estimation de densité.

4.4.5 Parallélisme

Un avantage très intéressant de notre méthode est qu'elle est aisément parallélisable. Notre implémentation dans l'outil SPECRAY permet d'effectuer le rendu sur une ferme de processeurs hétérogènes (on peut ainsi utiliser indifféremment un réseau de PC hétérogène ou une machine à multiples processeurs identiques). La seule condition étant d'avoir un disque partagé pour l'écriture de l'image finale. La distribution des tâches est effectuée à haut niveau, ce qui a l'avantage de ne pas nécessiter d'heuristique complexe pour la distribution des rayons et/ou des objets de la scène. La scène complète est chargée dans la mémoire de chaque calculateur. Un des nœuds de calcul, le distributeur, est mis en place pour la distribution des tâches aux autres nœuds. Chacun d'entre eux interroge le distributeur pour obtenir une passe à calculer. La seule information nécessaire est la graine d'initialisation des tirages aléatoires qui permet d'assurer que les cartes de photons générées sont distinctes pour tous les calculateurs. Chaque nœud génère sa propre carte et calcule l'image complète à partir de celle-ci. Une fois le traitement terminé, l'image de la passe est ajoutée à l'image finale. L'image finale est protégée par un système de jeton qui n'autorise l'écriture que d'un seul calculateur à la fois, les autres attendant si nécessaire que le jeton soit libéré. Enfin, lorsque le calcul est terminé, la luminance de l'image est divisée par le nombre de passes total. On stocke le nombre de passes générées afin de pouvoir reprendre le calcul si l'on souhaite augmenter la précision de l'image. La figure 4.12 illustre le principe du calcul sur une ferme de processeurs.

4.4.6 Algorithme

Pour résumer, nous présentons l'algorithme de notre lancer de photons multi-passes. Les algorithmes 1 et 2 présentent respectivement l'émission et la propagation des photons utilisées dans l'algorithme général 3. La figure 4.13 présente un exemple d'évolution de l'image finale au fil des passes. On constate que l'évolution de la précision au fil des passes est quadratique. Afin de diminuer l'erreur par deux, il faut multiplier le nombre de passes par deux. Cela permet d'avoir rapidement une bonne approximation de la solution.

4.5 Résultats

Notre méthode a été implémentée dans le logiciel SPECRAY de la société OKTAL Synthetic Environment (cf. annexe C.1). Les tests suivants ont été effectués sur un pentium cadencé à 2.4

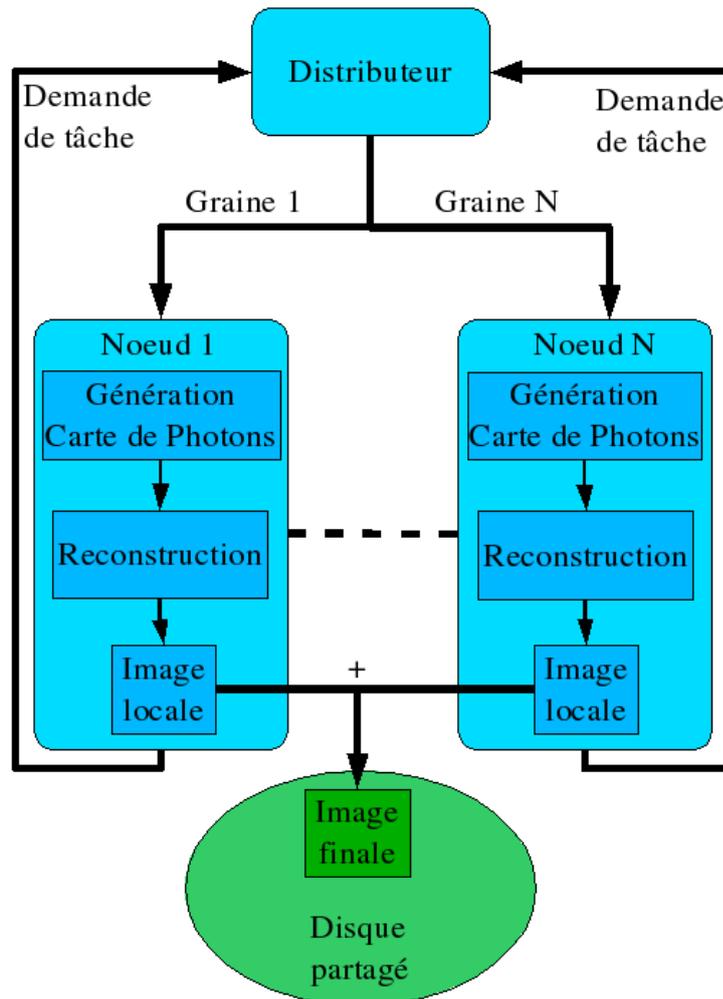


Figure 4.12. Principe du parallélisme du lancer de photons multi-passes.

GHz avec 512 Mo de mémoire. Le but ici est de démontrer la qualité obtenue par notre méthode, ainsi que la possibilité de son utilisation sur la plupart des systèmes.

4.5.1 Tube spéculaire

La figure 4.14 montre le résultat obtenu avec la nouvelle méthode pour le même tube émissif et spéculaire que dans la figure 4.8. On s'aperçoit que le rendu avec l'ancienne méthode était vraiment incorrect. Non seulement l'image était bruitée, mais les niveaux obtenus alors étaient

- Choisir la source lumineuse L
- Choisir la position d'émission sur L
- Choisir la direction d'émission en échantillonnant la fonction d'émission
- Affecter la puissance de la source L au photon p

Algorithme 1. Algorithme d'émission d'un photon.

```
Calculer l'intersection  $I_p$  du photon avec la scène
Mettre à jour la position du photon par  $I_p$ 
Stocker le photon dans la carte
Utiliser la Roulette Russe pour déterminer l'avenir du photon
Si le photon est absorbé alors
    Sortir
Fin si
Si le photon  $p$  est réfléchi alors
    Tirer la nouvelle direction du photon en échantillonnant par importance la FDRB
    Mette à jour la direction du photon
Fin si
```

Algorithme 2. *Algorithme de propagation d'un photon.*

erronés, ceci à cause du manque de photons pour l'estimation du pic de spécularité du matériau. La nouvelle méthode a permis de tracer autant de photons que c'était nécessaire. Dans le cas de la figure 4.14, on a pu lancer l'équivalent de 100 millions de photons sans augmenter la consommation mémoire. Le résultat obtenu est très satisfaisant, tant au niveau de la disparition du bruit dans l'image, que pour la précision obtenue.

4.5.2 Salle de conférence du Soda Hall

Ce test concerne le rendu, à la fois dans le spectre visible comme on le voit souvent et dans le spectre infrarouge, de la salle de Conférence du Soda Hall. La scène contient approximativement 170000 triangles. Dans le spectre visible, seuls 1300 triangles sont des sources étendus. En infrarouge, par contre, toute la scène est émissive. La figure 4.15 montre les images obtenues lors de ce test.

Le rendu visible a été effectué en 200 passes. Chaque passe utilise une carte de deux millions de photons dans laquelle on cherche 200 photons pour chaque estimation de densité. La taille de l'image générée est de 1024×768 pixels. Le rendu infrarouge a été effectué pour les trois bandes classiques en infrarouge décrite dans le chapitre 1. Ce rendu a été effectué en 100 passes. Chacune utilisant une carte d'un million de photons et cherchant 200 photons pour les estimations de densité. La taille des images est de 640×480 pixels.

Notre problématique ne porte pas sur la correction du biais sur les bords, ce problème n'a pas été implémenté dans notre méthode. Les artefacts dus à ce problème sont donc visibles sur

```

Initialiser le cache de luminance (image) à 0
Initialiser la graine des tirages aléatoires à 0
Pour chaque passe  $m$  faire
  #Calculer la carte de  $n$  photons
  Tant que le nombre de photons stockés est inférieur à  $n$  faire
    Emettre un photon
    Tant que le photon n'est pas absorbé faire
      Propager le photon à travers la scène
    Fin tant que
  Fin tant que
  Trier la carte de photons
  #Calculer l'image
  Pour chaque pixel  $p$  de l'image faire
    Calculer l'intersection  $I$  du rayon issu de l'oeil et passant par le pixel  $p$  avec la scène
    Rechercher les  $k$  plus proches photons du point  $I$ 
    Estimer la luminance au point  $I$  et la stocker dans le pixel.
  Fin pour
  Ajouter l'image calculée au cache de luminance.
Fin Pour
Diviser le cache de luminance par le nombre de passes  $m$  pour obtenir l'image finale.

```

Algorithme 3. *Algorithme du lancer de photons multi-passes.*

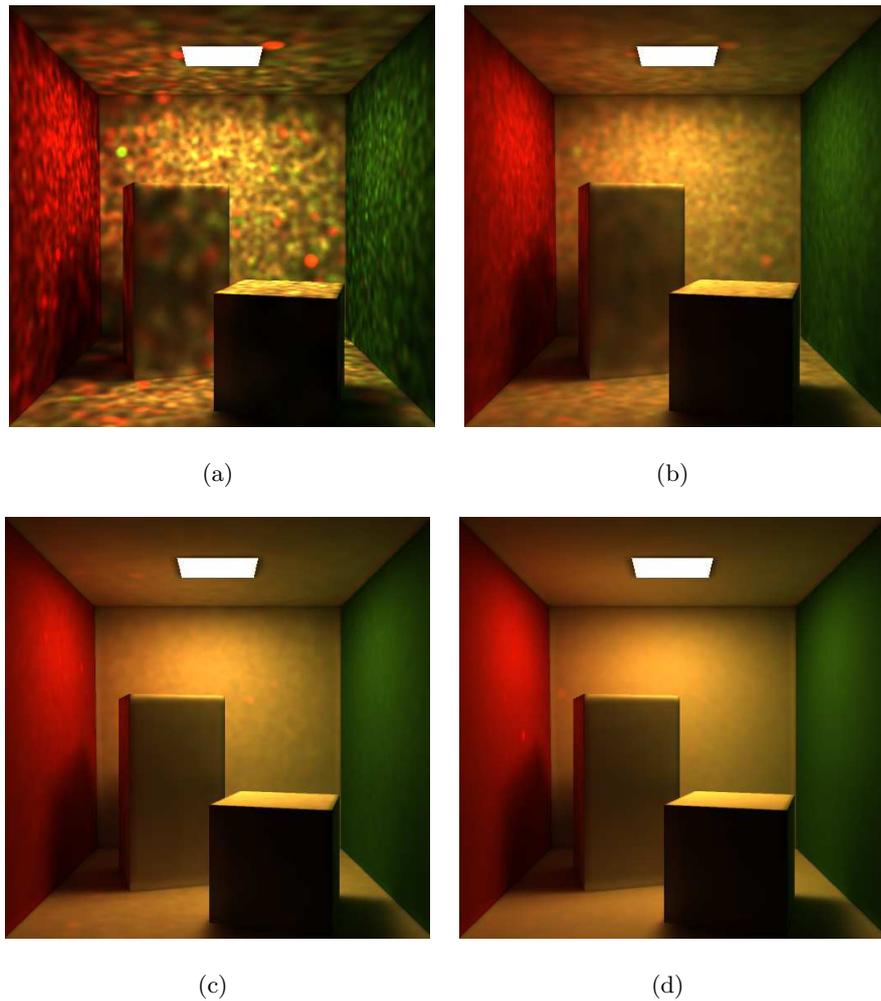


Figure 4.13. *Evolution du calcul multi-passes. (a) Première passe. (b) 2%. (c) 20%. (d) Image finale.*

les images générées (figure 4.15). Néanmoins il faut noter qu'on pourrait tout à fait intégrer dans notre méthode les stratégies de correction évoquées dans la section 3.2.3 de l'état de l'art.

4.5.3 Discussion

Nous calculons de petites cartes de photons. Toutes ces cartes sont globales, par opposition aux cartes locales dépendantes de la géométrie introduites dans [LC03]. Cela veut dire que chacune de nos cartes stocke des photons sur l'ensemble de la scène. Cela assure que l'illumination est répartie correctement dans la scène pour chaque passe de calcul et donc que les estimations de densité effectuées sont correctes. Nous bénéficions donc de l'accélération due à l'utilisation de multiples petites cartes de photons exposée dans [LC03] sans pour autant être limité par la

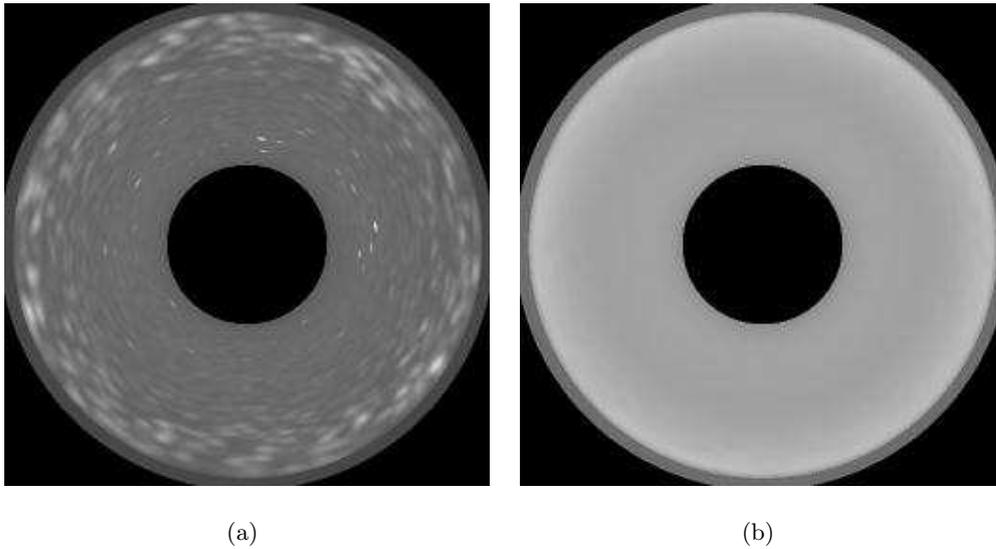


Figure 4.14. *Comparaison entre le lancer de photons spectral simple (a) et le lancer de photons multi-passes en utilisant cent million de photons (b) pour le rendu d'un tube émissif et spéculaire.*

complexité du maillage, car nos cartes ne sont pas dépendantes de la géométrie. Malheureusement, quand la taille de l'image devient grande (par exemple plus de 1024×768), le temps de calcul économisé sur le calcul et le tri de la carte de photons est perdu lors du rendu à cause du nombre d'estimations nécessaires. Il est donc plus intéressant de calculer moins de passes avec de plus grosses cartes lorsque l'on calcule de grandes images. La consommation mémoire étant constante d'une passe à l'autre.

Le cache que nous utilisons pour stocker l'illumination rend notre méthode dépendante de la taille de l'image, mais pas de la complexité de la scène comme dans [WRC88, SWH⁺95, Wal98] où le cache est dépendant de la régularité de l'illumination et/ou de la complexité géométrique. Notre méthode est donc plus adaptée pour le traitement de scènes complexes que [WRC88, SWH⁺95, Wal98], qui sont plus adaptées pour le traitement de scènes plus simples.

4.6 Conclusion et perspectives

Dans ce chapitre, nous avons présenté une méthode multi-passes d'estimation de densité capable d'obtenir une précision suffisante pour la simulation physique de l'infrarouge. Comme la méthode a été définie pour l'infrarouge, elle est capable de traiter toutes les surfaces de la scène comme des sources lumineuses. Dans le cadre du rendu visible, notre méthode est donc

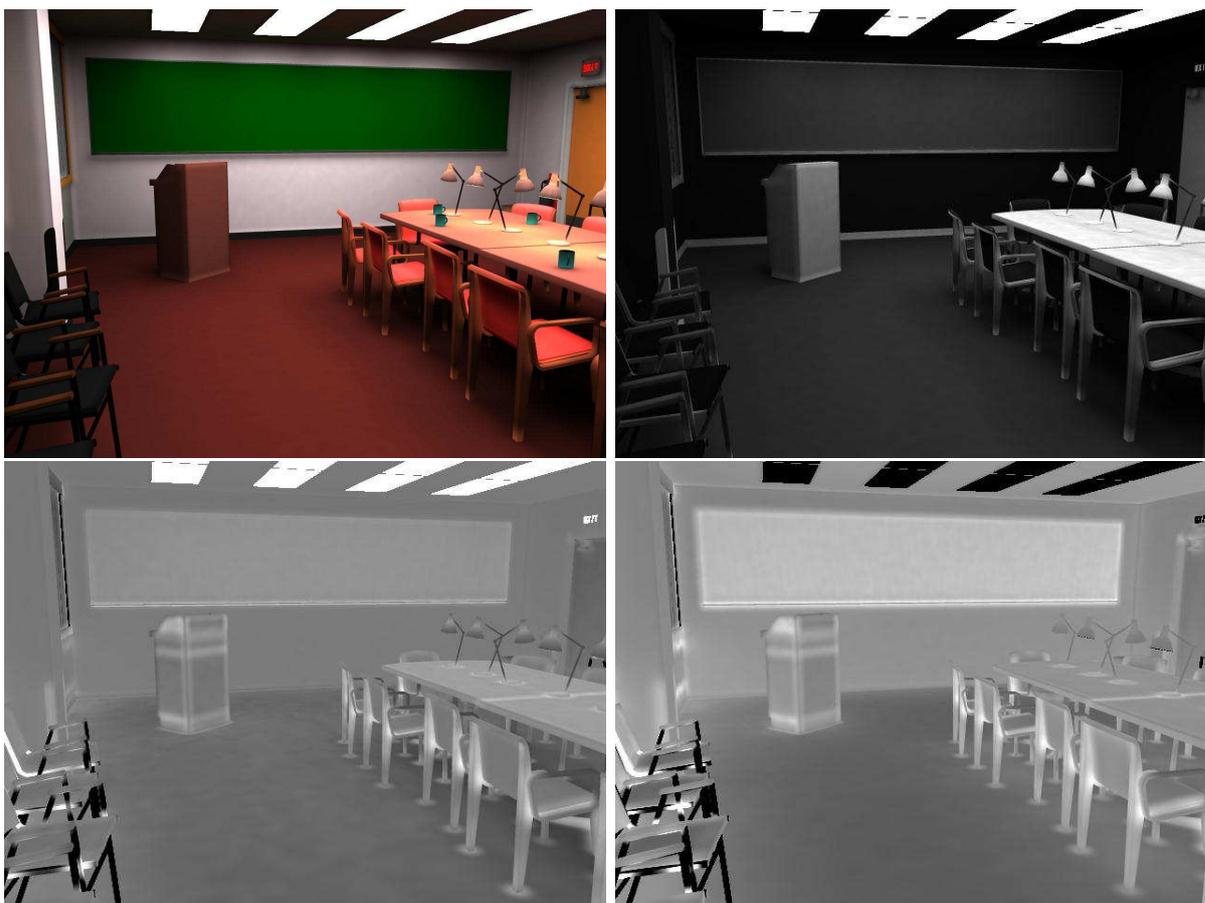


Figure 4.15. Rendu de la salle de conférence du Soda Hall en visible (en haut à gauche) et dans le spectre infrarouge : Bande I en haut à droite, Bande II en bas à gauche et Bande III en bas à droite.

très intéressante en ce qui concerne les scènes avec de nombreuses sources lumineuses. De plus notre méthode est indépendante de la complexité de la scène.

Bien entendu, la méthode peut encore être améliorée. La première des évolutions possibles serait l'implémentation d'un modèle spectral qui offre un coût de stockage moins élevé et des temps de calcul moins longs que le modèle général par bandes grises. Ensuite, notre méthode souffre de surestimation sur les bords. Il serait intéressant d'implémenter les solutions existantes pour ce problème [Jen01, LP03].

Afin de faire converger l'algorithme plus rapidement vers la solution, il serait judicieux d'effectuer un traitement particulier pour les matériaux fortement spéculaires. Par exemple, on pourrait utiliser la visualisation directe de la carte de photons associée à un échantillonnage du lobe spéculaire par des rayons supplémentaires. Nous pourrions également analyser la carte de

photons calculée à chaque passe afin d'en tirer une information d'importance qui permettrait de guider la construction de la carte suivante, et ainsi réduire le nombre de passes nécessaires pour obtenir la même qualité. De plus, les recherches des photons pourraient être accélérées en utilisant une meilleure structure de données pour la carte. En effet, le *kd-tree* proposé par Jensen offre un bon compromis entre rapidité de recherche et consommation mémoire. Or nous ne sommes plus limités par la consommation mémoire, et dans ce cas, il existe des structures qui permettent des recherches plus rapides.

Enfin, notre méthode traite les milieux participants en utilisant un *Ray-Marching* adaptatif comme dans [Jen01]. Cette méthode offre de bons résultats qualitatifs. Néanmoins, cela est très coûteux en temps de calcul. Dans le chapitre 5, nous présentons une évolution de la méthode qui minimise le temps de rendu pour ces milieux.

Chapitre 5

Ecrasement de photons pour les milieux participants

5.1 Introduction

Comme dans le cas des surfaces, la méthode des cartes de photons est actuellement la plus efficace en ce qui concerne le calcul de l'illumination globale dans les milieux participants. Malgré tout, les temps de rendu restent très importants. Dans la section 5.2, nous décrivons le principe de la méthode et mettons en lumière les problèmes de l'algorithme. Ensuite, nous montrons qu'une reformulation de l'équation volumique du transfert radiatif permet de séparer le calcul des différents phénomènes (section 5.3). Nous proposons une méthode qui combine un ray marching pour le calcul de l'émission, de l'absorption et de la diffusion sortante (section 5.5) et une passe d'estimation de densité pour le calcul de la diffusion entrante (section 5.6). La méthode présentée dans ce chapitre a été publiée dans l'article [BPPP05].

5.2 Discussion de la méthode des cartes de photons pour les milieux participants

Comme nous l'avons vu au chapitre 3, la méthode des cartes de photons est une méthode en deux passes. La première passe consiste à propager les photons depuis les sources à travers le milieu participant. La différence entre notre méthode et celle de Jensen concernant la construction de la carte de photons étant minime, nous nous contentons de décrire cette phase pour notre

méthode (section 5.6.3.1). La seconde passe consiste à reconstruire la luminance des rayons issus des pixels en effectuant un ray marching. Jensen découpe le rayon en petits segments sur lesquels les propriétés du milieu sont supposées constantes. Il évalue alors la luminance émise par le milieu sur la longueur du segment ainsi que l'absorption et la diffusion sortante. Il tire également un point aléatoirement sur le segment et évalue la diffusion entrante en ce point par une recherche des k plus proches voisins. Les problèmes de performances de la méthode proviennent de ce dernier point.



Figure 5.1. Illustration du bruit poivre et sel dû au manque d'échantillons.

Si l'on considère une carte de photon, implémentée par un *kd-tree* [Ben79], contenant N photons, la complexité de recherche des k plus proches photons est $O(k + \log N)$. Le problème est qu'ici, contrairement au cas des surfaces, on doit effectuer un nombre de recherche égal au nombre de segments pour chaque rayon. Or, la méthode étant stochastique, elle converge vers la solution quand le nombre d'échantillons tend vers l'infini, c'est-à-dire quand le nombre de segments tend vers l'infini. Ce nombre peut avoir une forte dépendance sur la variation du milieu, surtout dans le cas d'un ray marching adaptatif. La figure 5.1 illustre le problème de bruit poivre et sel dû au manque d'échantillons. Pour un rayon discrétisé en n segments, la complexité de l'algorithme est donc de $O(n(k + \log N))$. De plus, comme le montre la figure 5.2, deux évaluations effectuées pour deux segments voisins peuvent utiliser les mêmes photons ce qui cause des recherches redondantes dans la carte de photons.

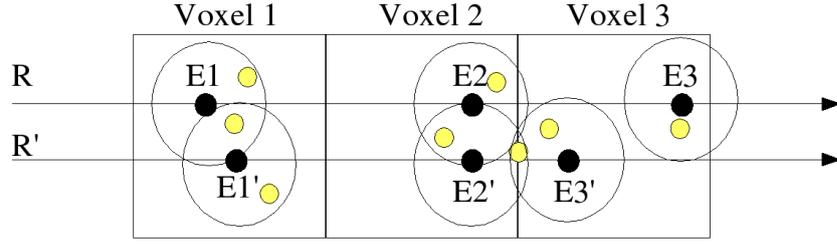


Figure 5.2. Illustration de la redondance des recherches par la méthode des cartes de photons.

5.3 Formulation itérative de l'équation volumique du transfert radiatif

Reprenons l'expression intégrée de l'équation volumique du transfert radiatif (équation 2.12) :

$$L(x, \omega) = e^{-\tau(x_0, x_n)} L(x_0, \omega) + \int_{x_0}^{x_n} e^{-\tau(u, x_n)} \sigma_a(u) L_e(u, \omega) du \\ + \int_{x_0}^{x_n} e^{-\tau(u, x_n)} \frac{\sigma_s(u)}{4\pi} \int_{\Omega=4\pi} p(u, \omega, \omega') L(u, \omega') d\omega' du$$

Cette équation peut être résolue par une intégration numérique. Cette intégration consiste à couper le rayon en petits segments sur lesquels tous les coefficients sont considérés comme constants. Pour simplifier les notations, nous appelons $L_\omega(u)$, la luminance entrante provenant de toutes les directions au point u :

$$L_\omega(u) = \int_{\Omega=4\pi} p(u, \omega, \omega') L(u, \omega') d\omega' \quad (5.1)$$

En utilisant cette notation, nous pouvons réécrire l'équation du transfert radiatif de manière récursive en exprimant la luminance en x_n en fonction de la luminance x_{n-1} :

$$L(x_n) = e^{-\tau(x_{n-1}, x_n)} L(x_{n-1}) \\ + \int_{x_{n-1}}^{x_n} e^{-\tau(u, x_n)} \sigma_a(u) L_e(u) du \\ + \int_{x_{n-1}}^{x_n} e^{-\tau(u, x_n)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \quad (5.2)$$

Nous omettons la direction ω dans les notations pour simplifier l'expression. Néanmoins, tous les termes de luminance où elle est omise en dépendent. On s'intéresse en effet à la luminance le long du rayon d'observation de direction ω .

La luminance au pas x_n est fonction de la luminance au pas x_{n-1} . En effet, $L(x_n)$ dépend de toutes les diffusions ayant lieu dans les pas précédents. Mais, notre idée est d'utiliser deux différentes méthodes pour traiter d'une part l'émission, l'absorption et la diffusion sortante,

et d'autre part la diffusion sortante. Nous cherchons donc à reformuler l'équation récursive du transfert radiatif en une équation itérative qui nous permettra de calculer chaque terme séparément.

Nous prenons comme condition initiale de la relation par récurrence la luminance $L(x_0)$ à la sortie du milieu. Nous pouvons exprimer la luminance $L(x_1)$ en fonction de cette condition initiale :

$$\begin{aligned} L(x_1) &= e^{-\tau(x_0, x_1)} L(x_0) \\ &+ \int_{x_0}^{x_1} e^{-\tau(u, x_1)} \sigma_a(u) L_e(u) du \\ &+ \int_{x_0}^{x_1} e^{-\tau(u, x_1)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \end{aligned} \quad (5.3)$$

$$(5.4)$$

De même nous pouvons exprimer $L(x_2)$ en fonction de $L(x_1)$ et donc par extension en fonction de $L(x_0)$:

$$\begin{aligned} L(x_2) &= e^{-\tau(x_1, x_2)} e^{-\tau(x_0, x_1)} L(x_0) \\ &+ e^{-\tau(x_1, x_2)} \int_{x_0}^{x_1} e^{-\tau(u, x_1)} \sigma_a(u) L_e(u) du \\ &+ e^{-\tau(x_1, x_2)} \int_{x_0}^{x_1} e^{-\tau(u, x_1)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \\ &+ \int_{x_1}^{x_2} e^{-\tau(u, x_2)} \sigma_a(u) L_e(u) du \\ &+ \int_{x_1}^{x_2} e^{-\tau(u, x_2)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \end{aligned} \quad (5.5)$$

$$(5.6)$$

On peut finalement obtenir l'expression de la luminance $L(x_n)$ à l'entrée du milieu en fonction de la luminance en sortie :

$$\begin{aligned} L(x_n) &= \prod_{i=0}^{n-1} e^{-\tau(x_i, x_{i+1})} L(x_0) \\ &+ \sum_{j=1}^{n-1} \prod_{i=j}^{n-1} e^{-\tau(x_i, x_{i+1})} \int_{x_{j-1}}^{x_j} e^{-\tau(u, x_j)} \sigma_a(u) L_e(u) du \\ &+ \sum_{j=1}^{n-1} \prod_{i=j}^{n-1} e^{-\tau(x_i, x_{i+1})} \int_{x_{j-1}}^{x_j} e^{-\tau(u, x_j)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \\ &+ \int_{x_{n-1}}^{x_n} e^{-\tau(u, x_n)} \sigma_a(u) L_e(u) du \\ &+ \int_{x_{n-1}}^{x_n} e^{-\tau(u, x_n)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \end{aligned} \quad (5.7)$$

Afin de simplifier l'expression, on pose :

$$P(j) = \begin{cases} 1 & \text{si } j = n, \\ \prod_{i=j}^{n-1} e^{-\tau(x_i, x_{i+1})} & \text{sinon.} \end{cases}$$

On obtient alors une expression itérative de la luminance au point d'entrée :

$$\begin{aligned} L(x_n) &= \prod_{i=0}^{n-1} e^{-\tau(x_i, x_{i+1})} L(x_0) \\ &+ \sum_{j=1}^n P(j) \int_{x_{j-1}}^{x_j} e^{-\tau(u, x_j)} \sigma_a(u) L_e(u) du \\ &+ \sum_{j=1}^n P(j) \int_{x_{j-1}}^{x_j} e^{-\tau(u, x_j)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \end{aligned} \quad (5.8)$$

L'intérêt de cette formulation est qu'elle est composée d'une somme de termes indépendants les uns des autres, et qui peuvent donc être calculés séparément. Ces trois termes correspondent respectivement à :

- la luminance en sortie atténuée par l'absorption et à la diffusion sortante,
- la luminance émise par le milieu sur le trajet du rayon dans le milieu,
- et la luminance due à la diffusion entrante sur ce même trajet.

Nous utilisons deux méthodes différentes pour calculer la luminance. La section 5.5 expose la méthode que nous utilisons pour calculer la contribution de l'émission, l'absorption et la diffusion sortante. La section 5.6.3 présente notre méthode pour le calcul de la diffusion entrante.

5.4 Description du milieu

Comme nous l'avons vu dans le chapitre 2, un certain nombre de paramètres sont utilisés pour décrire l'interaction de la lumière avec le milieu. Nous utilisons une grille régulière de voxels pour décrire le milieu. Pour chaque voxel, les données sont supposées constantes. Ces données sont :

- le coefficient d'absorption, σ_a ,
- le coefficient de diffusion, σ_s ,
- la température du voxel, utilisée pour calculer la luminance émise grâce à la loi du corps noir,
- le coefficient d'asymétrie g , permettant de paramétrer la fonction de phase de Heyney-Greenstein que nous utilisons pour décrire la diffusion.

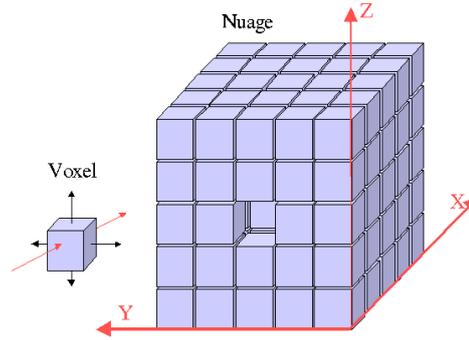


Figure 5.3. Structure de données utilisée pour représenter le milieu participant.

L'utilisation d'une grille régulière a l'avantage de proposer l'accès le plus rapide aux données stockées. Néanmoins, la consommation mémoire d'une telle structure devient rapidement importante. Il serait donc préférable d'utiliser une structure adaptative moins gourmande (i.e. un *octree*). Heureusement, la méthode que nous proposons s'adapte facilement à toutes les structures de données.

5.5 Simulation de l'émission, de l'absorption et de la diffusion sortante

Considérons l'émission, l'absorption et la diffusion sortante dans l'équation 5.8. On note $L_{RM}(x_n)$ la luminance due à ces phénomènes. La découpe du rayon est effectuée de manière à ce que les propriétés du milieu soient constantes sur chaque segment et à ce que l'épaisseur optique soit constante et proche de 0 sur chaque segment (i. e. Ray Marching adaptatif). Dans ce cas, $e^{-\tau(u,x)} \approx 1$. Cela signifie que la luminance est calculée en supposant qu'elle est constante sur le segment. Grâce à ces hypothèses, nous pouvons exprimer $L_{RM}(x_n)$ de manière discrète :

$$L_{RM}(x_n) = \prod_{i=0}^{n-1} e^{-\sigma_t(x_i)\Delta x_i} L(x_0) + \sum_{j=1}^n P(j)\sigma_a(x_j)L_e(x_j)\Delta x_j \quad (5.9)$$

et

$$P(j) = \begin{cases} 1 & \text{si } j = n, \\ \prod_{i=j}^{n-1} e^{-\sigma_t(x_i)\Delta x_i} & \text{sinon.} \end{cases}$$

où $\Delta x_i = \|x_i, x_{i+1}\|$ est la longueur de chaque segment.

Cette équation est évaluée par une méthode de type Ray Marching dont le principe est illustré par la figure 5.4. La luminance est stockée pour chaque rayon, de manière à pouvoir la

combiner avec la diffusion sortante à l'issue du calcul.

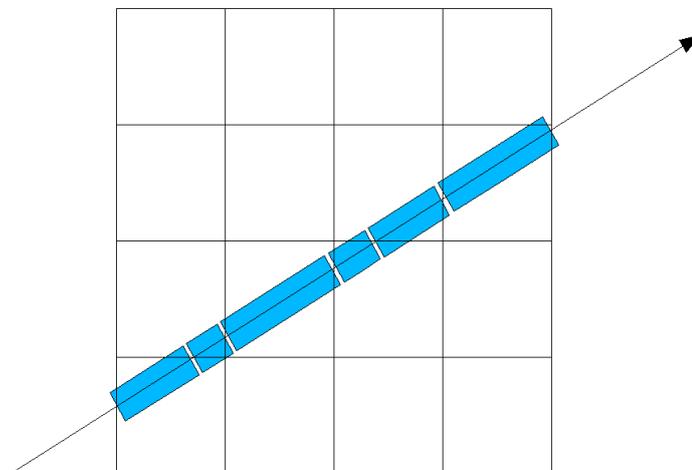


Figure 5.4. La méthode du ray marching calcule la contribution du milieu en découpant le rayon en petits segments. Pour chaque segment, le milieu est supposé homogène. Par conséquent, dans le cas d'une grille régulière, les petits segments correspondent au parcours du rayon dans chaque voxel traversé.

5.6 Simulation de la diffusion entrante

Nous décrivons tout d'abord la méthode de l'écrasement de photons [LP03] utilisée pour l'illumination globale de scènes sans milieux participants. Après un inventaire des méthodes utilisées en rendu volumique, nous présentons une méthode du calcul de la diffusion entrante qui conjugue rendu volumique et écrasement de photons pour la visualisation de la carte de photons.

5.6.1 Description de l'écrasement de photons

L'écrasement de photons est une méthode d'estimation de densité duale qui a été proposée pour optimiser la méthode des cartes de photons. Le principe de la méthode reste le même, à savoir le précalcul de l'illumination par propagation des photons dans la scène, puis la reconstruction de la luminance par estimation de densité.

Dans l'écrasement de photons, la construction de la carte de photons est similaire à celle que nous décrivons dans le chapitre 3. La différence provient de la méthode de reconstruction. En effet, dans la méthode originale [Jen01], Jensen utilise la recherche des k plus proches voisins pour sélectionner les photons utilisés pour l'estimation de la luminance. Ce qui constitue une

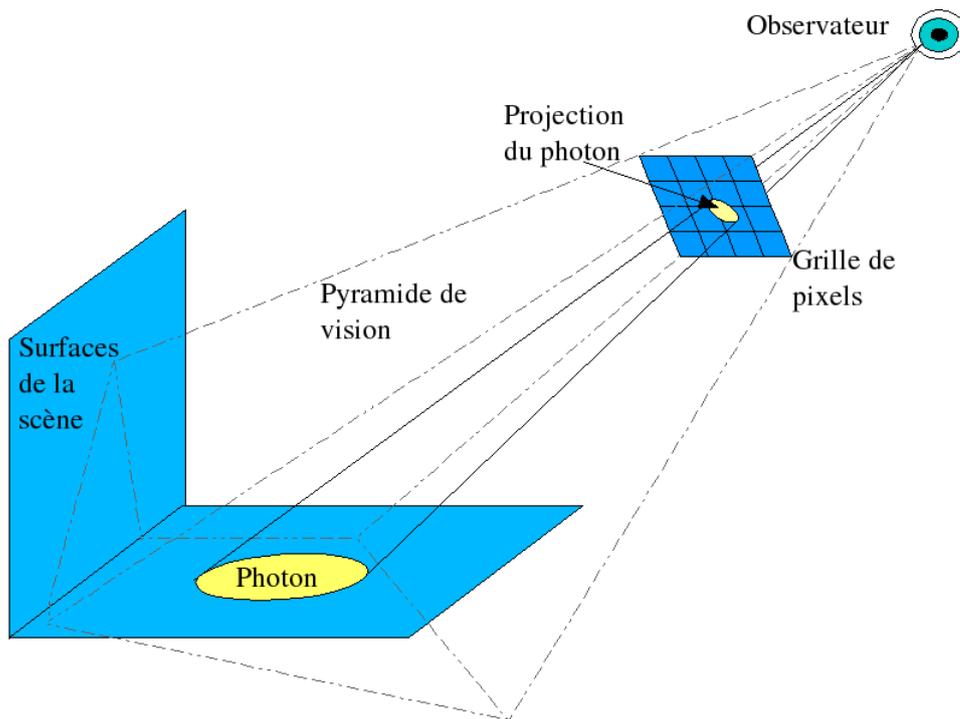


Figure 5.5. Principe de l'écrasement de photons. Le disque associé au photon est projeté sur la grille de pixels de l'écran.

approche directe de l'estimation de densité. Lavignotte préfère utiliser une approche duale de l'estimation de densité. C'est-à-dire sélectionner pour chaque photon les points d'estimation susceptibles de recevoir une contribution non nulle pour ce photon. Afin de se détacher de la complexité de la scène, il choisit comme points d'estimation les pixels de l'image à générer. En effet, la reconstruction de la luminance ne s'effectue plus dans l'espace scène, mais dans l'espace image. Comme on le présente sur la figure 5.5, la sélection des pixels influencés par un photon est effectué par projection sur l'écran d'un disque centré sur le photon et orienté selon la normale de la surface à laquelle il appartient. Les pixels recouverts sont les points d'estimation pour lesquels la contribution du photon est non nulle. Il suffit donc d'ajouter à ces pixels la puissance du photon modulée par la FDRB et par le poids du noyau choisi.

Un des avantages de cette approche est que la projection du photon sur l'écran est un problème classique de rasterisation d'un disque (projection du disque sur la grille de pixels). L'intérêt est que ceci peut être fait en utilisant les capacités des cartes graphiques qui sont parfaitement adaptées pour ce genre de traitement. En effet, le photon peut être représenté par un quadrilatère texturé par la fonction noyau discrétisée sous la forme d'un disque. La couleur associée au quadrilatère correspond au poids RVB du photon. Ce quadrilatère est ensuite

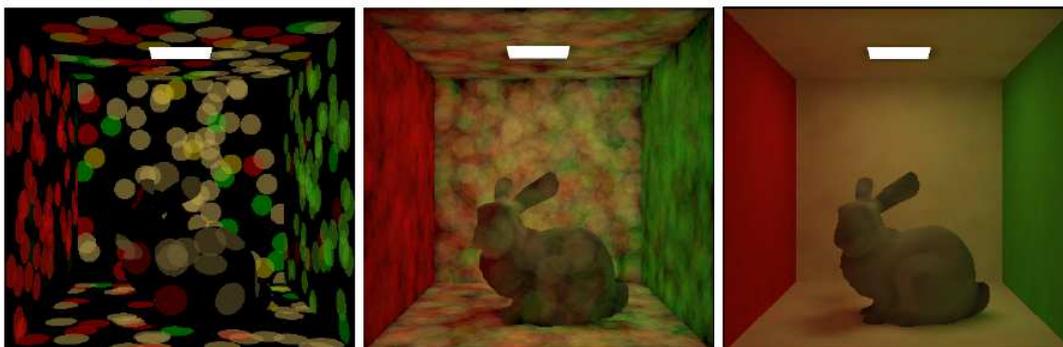


Figure 5.6. Exemple de l'évolution de l'écrasement de photons. De gauche à droite : 0,05%, 1%, 100% du calcul.

rendu par la bibliothèque graphique OpenGL en utilisant les capacités de mélange additif. La figure 5.6 présente un exemple de l'évolution du calcul. A l'époque où Lavignotte propose cette méthode, les capacités des cartes graphiques étaient limitées et commençaient seulement à être programmables. Il était donc obligé de résoudre les contraintes sur la précision des calculs. En effet ceux-ci s'effectuaient en utilisant seulement 8 bits par canal. Pour régler ces problèmes il aurait fallu effectuer le rendu dans un tampon de flottants, mais le mélange n'était pas encore implémenté pour ce type de tampons. Il propose donc une implémentation qui permet d'effectuer le rendu dans un tampon d'entiers qui consiste à compter les photons qui se projettent sur un pixel. Bien que fonctionnelle, cette implémentation est aujourd'hui inutile, car les nouvelles générations de cartes graphiques permettent d'effectuer du rendu directement en flottant. Cela facilite grandement l'implémentation de la méthode et augmente d'autant plus son intérêt qu'on se débarrasse ainsi de la contrainte qu'il a introduit concernant l'équité de la puissance des photons. En effet, pour que cette contrainte soit respectée, il était nécessaire de modifier l'algorithme de propagation. On peut donc maintenant conserver l'algorithme de propagation classique.

5.6.2 Rendu volumique

Le rendu volumique est une technique importante en ce qui concerne la visualisation de données mesurées ou simulées dans des applications scientifiques et industrielles. On peut citer par exemple l'imagerie médicale pour la visualisation de données mesurées au scanner. Le rendu volumique idéal reconstruit une fonction 3D continue, transforme cette fonction 3D vers l'espace image, et évalue les intégrales d'opacité le long des lignes de vue. En 1989, Westover [Wes89,

Wes90] introduit l'*écrasement* (i.e. *splatting* en anglais) pour la visualisation interactive, ce qui constitue une approximation du rendu volumique. Les algorithmes d'écrasement interprètent les données volumiques comme des ensembles de particules qui absorbent et émettent de la lumière. Les intégrales linéaires sont précalculées au travers de chaque particule séparément, créant ainsi des fonctions d'empreinte (i.e. *footprint functions* en anglais). Chaque empreinte répartit sa contribution dans le plan image. Ces contributions sont composées de l'arrière vers l'avant dans l'image finale.

On voit ici se profiler l'intérêt que nous portons à ces méthodes. Il est aisé de faire l'association entre l'écrasement de photons et l'écrasement volumique. On peut en effet considérer la carte de photon comme un ensemble de données volumiques. Les méthodes d'écrasement ayant déjà été appliquées au rendu des surfaces [SB97, LP03], il semble logique d'étendre la méthode aux cartes de photons volumiques.

Les premiers algorithmes d'écrasement [Wes89] souffraient de la détermination imprécise de la visibilité lorsque les *splats* (i.e. volumes écrasés) sont composés de l'arrière vers l'avant. Cela cause des artefacts visibles comme le débordement des couleurs. Plus tard, Westover [Wes90] a résolu ce problème en utilisant un tampon d'écrasement aligné avec les axes du volume. Cependant, cette technique provoque des artefacts clignotants lors de l'animation. Pour résoudre ce problème, Mueller [MMI⁺98] proposa d'aligner les tampons d'écrasement avec l'image plutôt qu'avec les données volumiques. De plus, ils écrasent chaque couche du noyau de reconstruction séparément. Cette méthode est très similaire à [CCF94, GK96]. Mueller et Yagel [MY96] combinent l'écrasement au lancer de rayon pour accélérer le rendu en projection perspective. Laur et Hanrahan [LH91] décrivent un algorithme d'écrasement hiérarchique qui permet un raffinement progressif de la solution pendant le rendu. Un grand nombre de noyaux de reconstruction différents (i.e. empreintes) ont été proposés, et notamment par Zwicker *et al.* [ZPVG01] qui étendent les travaux d'Heckbert pour le rendu par point [Hec89]. Cette méthode est nommée écrasement de volume EWA (i.e. *elliptical weighted average*). Le rendu volumique EWA est très attractif, car il évite les artefacts dus à l'anti-crénelage dans l'image générée ainsi que les flous excessifs. Cette méthode a été ensuite portée sur les processeurs graphiques par Chen [CDK04] pour améliorer les performances.

En nous inspirant de ces travaux et de la méthode de l'écrasement de photons, nous allons proposer une méthode d'écrasement de photons volumique qui minimise les temps de calcul par rapport à la méthode des cartes de photons.

5.6.3 Principe de l'écrasement de photons volumique

Comme nous l'avons vu dans le chapitre 3, la méthode des cartes de photons [Jen01] est la plus intéressante pour le rendu de milieux participants. Par conséquent, nous essayons de garder le même principe de calcul en deux passes.

5.6.3.1 Emission et propagation

Concernant la propagation du flux, nous avons fait évoluer la méthode spectrale présentée au chapitre 4 de la même façon que Jensen a fait évoluer sa méthode originale pour l'adaptation aux milieux participants.

Emission L'émission des photons est très similaire à la différence près qu'on introduit les voxels du milieu en tant que sources potentielles pour le cas de milieux émissifs (comme le feu en visible).

Propagation La propagation par contre est plus complexe. En effet, il faut assurer une propagation correcte des photons dans le milieu en fonction de ses caractéristiques.

Les photons sont tracés par lancer de rayons à travers les voxels formant le nuage, prenant en compte, le cas échéant, le masquage des objets interceptés par le nuage. Dans le cas où un photon traverse un milieu participatif homogène, la distance moyenne d'interaction avec le milieu est :

$$\bar{d}_h = \frac{1}{\bar{\sigma}_t}$$

où $\bar{\sigma}_t$ est la valeur intégrée sur le domaine spectral du coefficient d'extinction :

$$\bar{\sigma}_t = \frac{1}{\Delta\Lambda} \int_{\Delta\Lambda} \sigma_{t_\lambda} d\lambda$$

Le principe est le même que dans le chapitre 4 pour la réflectance.

Dans le cas où le milieu traversé est non homogène, la distance d'interaction moyenne devient :

$$\bar{d} = \frac{1}{\tau(0, \bar{d}_h)}$$

avec :

$$\tau(0, \bar{d}_h) = \int_x^{x+\bar{d}_h} \bar{\sigma}_t(u) du$$

τ est l'épaisseur optique du milieu sur le trajet $[x, x + \bar{d}_h]$. Celle-ci est évaluée par Ray Marching. On effectue l'intégration sur les données discrètes représentant le milieu. Si la distance

\overline{d}_h correspond à la traversée de n voxels, on découpe le segment $[x, x + \overline{d}_h]$ en n segments. On a donc :

$$\tau(0, \overline{d}_h) \approx \sum_{i=1}^n \overline{\sigma}_t(i) d_i$$

où $\overline{\sigma}_t(i)$ est le coefficient d'extinction intégré du i -ème voxel et d_i la distance parcourue dans ce voxel. Pour améliorer la qualité de l'échantillonnage, il est préférable d'utiliser une distance d'interaction aléatoire :

$$\Delta l = -\frac{\ln \xi}{\tau(0, \overline{d}_h)}$$

où ξ est un nombre aléatoire tiré uniformément entre 0 et 1. L'intérêt de ce tirage est que la valeur moyenne des distances tirées est \overline{d} .

Une fois déterminé le point d'interaction, le photon est stocké et on utilise une version de la Roulette Russe adaptée aux milieux participants. Celle-ci se base sur l'albédo de diffusion :

$$\Lambda = \frac{\overline{\sigma}_s}{\overline{\sigma}_t}$$

On tire donc un nombre aléatoire ξ uniformément entre 0 et 1 :

- si $\xi \leq \Lambda$ alors le photon est diffusé
- sinon le photon est absorbé.

Si le photon est absorbé, un nouveau est émis si nécessaire. S'il est diffusé, la nouvelle direction de propagation est choisie en échantillonnant la fonction de phase par importance. Nous avons choisi d'utiliser la fonction d'Henye-Greenstein car elle a l'avantage de pouvoir être échantillonnée grâce à un tirage par importance de manière analytique. En effet, un tirage par importance de l'angle θ en fonction de cette fonction est donné par :

$$\cos \theta = \frac{1}{2g} \left(1 + g^2 - \left(\frac{1 - g^2}{1 - g + 2g\xi} \right)^2 \right) \quad (5.10)$$

où ξ est un nombre aléatoire tiré uniformément entre 0 et 1.

De même que dans le cas des surfaces, nous devons prendre en compte le fait que l'on raisonne sur des coefficients intégrés au lieu des coefficients spectraux. On doit donc modifier le poids du photon en conséquence. La fonction de densité de probabilité pour une diffusion est :

$$FDP_d(\Delta\lambda) = \overline{\sigma}_s$$

Pour chaque bande $\Delta\lambda$ le poids du photon devient :

$$\Phi_{d,\Delta\lambda} = \Phi_{i,\Delta\lambda} \frac{\sigma_{s,\Delta\lambda}}{FDP_d(\Delta\lambda)}$$

où $\Phi_{i,\Delta\lambda}$ est la puissance du photon incident pour la bande $\Delta\lambda$ et $\Phi_{d,\Delta\lambda}$ est la puissance du photon diffusé.

5.6.3.2 Prise en compte de l'illumination directe

Comme dans le cas des surfaces, la méthode de Jensen [Jen01] traite explicitement l'illumination directe. Cependant, comme dans le chapitre 4, nous voulons traiter autant de sources étendues que nécessaire. Or tracer suffisamment de rayons d'ombres vers les sources étendues de façon à réduire le bruit est plus coûteux que simplement tracer plus de photons. D'autant plus que dans le cas des milieux participants, on doit effectuer un Ray Marching pour chaque rayon d'ombre. C'est pourquoi nous avons choisi de traiter l'ensemble de l'illumination d'un seul coup par estimation de densité de la carte de photons. Nous stockons donc toutes les interactions de la lumière avec le milieu, même les primaires. De cette manière, l'illumination directe est prise en compte automatiquement par l'évaluation de la carte de photons. De plus, l'évaluation directe dans le cas des milieux participants est moins gênante que pour les surfaces. En effet, le bruit basse fréquence est moins visible car ces milieux ont un caractère beaucoup plus lisse que les surfaces.

5.6.3.3 Stockage de la carte de photons

Dans notre méthode, la recherche des photons n'est plus nécessaire. Par contre on doit traiter les photons séquentiellement, mais selon la propriété de l'observateur (champ, direction d'observation...) tous les photons ne sont pas forcément visibles lors du calcul de l'image. Dans un souci d'optimisation il semble donc intéressant de déterminer les photons qui appartiennent à la pyramide de vision. Nous avons donc changé la structure de données de la carte de photon pour préférer un *octree* au *kd-tree* utilisé dans la méthode du chapitre 4, structure qui est plus adaptée aux nouveaux besoins.

5.6.3.4 Reconstruction

La reformulation de l'équation du transfert radiatif sous une forme itérative nous permet de traiter séparément la contribution de la diffusion entrante. Ceci est très intéressant car il s'agit de la partie coûteuse dans la méthode des cartes de photons classique.

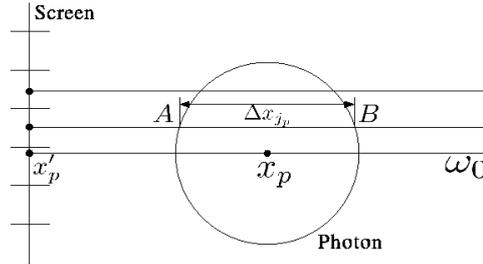


Figure 5.7. Géométrie associée à l'écrasement de photons.

On note $L_{IS}(x_n)$ la luminance due à la diffusion entrante. On a :

$$L_{IS}(x_n) = \sum_{j=1}^n P(j) \int_{x_{j-1}}^{x_j} e^{-\tau(u, x_j)} \frac{\sigma_s(u)}{4\pi} L_\omega(u) du \quad (5.11)$$

En utilisant les mêmes hypothèses simplificatrices que pour l'équation 5.9, on obtient :

$$L_{IS}(x_n) = \sum_{j=1}^n P(j) \frac{\sigma_s(x_j)}{4\pi} \Delta x_j L_\omega(x_j) \quad (5.12)$$

On choisit de calculer $L_\omega(x_j)$ par estimation de densité de la carte de photons. Considérons la relation entre le flux et la luminance dans les milieux participants :

$$L(x, \omega) = \frac{d^2\Phi(x, \omega)}{\sigma_s(x) d\omega dV} \quad (5.13)$$

où dV est un élément de volume. On peut maintenant intégrer l'équation 5.1 en utilisant la précédente relation et l'estimation de densité. On obtient :

$$L_\omega(x_j) = \sum_{p=1}^N \left[K_h(x_j - x_p) p(x_j, \omega_0, \omega_p) \frac{\Delta\phi(x_p, \omega_p)}{\sigma_s(x_j) \Delta V} \right] \quad (5.14)$$

avec N le nombre total de photons, K_h la fonction noyau, x_p la position du photon, ω_p la direction d'incidence du photon et ΔV le petit volume contenant les photons utilisés pour l'estimation.

En remplaçant L_ω dans l'équation 5.12, on a :

$$L_{IS}(x_n) = \sum_{j=1}^n P(j) \frac{\sigma_s(x_j)}{4\pi} \Delta x_j \sum_{p=1}^N K_h(x_j - x_p) p(x_j, \omega_0, \omega_p) \frac{\Delta\phi(x_p, \omega_p)}{\sigma_s(x_j) \Delta V}$$

On peut intervertir les sommes dans l'équation 5.15 :

$$L_{IS}(x_n) = \sum_{p=1}^N \sum_{j=1}^n \left[P(j) \frac{1}{4\pi} \Delta x_j K_h(x_j - x_p) p(x_j, \omega_0, \omega_p) \frac{\Delta\phi(x_p, \omega_p)}{\Delta V} \right] \quad (5.15)$$

Or, dans l'équation 5.15, on remarque que pour chaque photon p , il n'y a qu'un petit nombre de termes de la somme sur j qui sont non nuls. En effet, la valeur de la fonction noyau est égale

à zéro pour tous les segments qui sont plus loin que la taille du support du noyau. On peut donc remplacer la somme sur j par un simple terme non nul d'index noté j_p . Finalement, on obtient :

$$L_{IS}(x_n) = \frac{1}{4\pi} \sum_{p=1}^N \left[P(j_p) p(x_{j_p}, \omega_0, \omega_p) \frac{K_h(x_{j_p} - x_p) \Delta x_{j_p}}{\Delta V} \Delta \phi(x_p, \omega_p) \right] \quad (5.16)$$

L'équation 5.16 donne la luminance d'un rayon correspondant à un pixel de l'écran. Comme nous voulons effectuer l'estimation dans l'espace image, nous n'avons pas d'information sur le point x_{j_p} . Mais, nous voyons que le terme Δx_{j_p} correspond à la portion du rayon qui est influencé par le photon (figure 5.7). De plus, la fonction noyau caractérise la façon dont le photon influence le rayon. Nous intégrons donc la fonction noyau sur le chemin Δx_p du rayon à travers le photon de manière à connaître la contribution du photon pour ce rayon. Pour les autres termes qui utilisent x_{j_p} , on pose $x_{j_p} \approx x_p$.

Similarités avec le rendu volumique Pour chaque photon, nous devons évaluer la contribution qu'il apporte à tous les rayons issus des pixels de l'écran qui le traversent. La première idée serait de calculer séquentiellement cette contribution. Néanmoins, cela serait fastidieux. Or le photon peut être considéré comme une donnée volumique. Parmi les méthodes de rendu volumique, nous trouvons les méthodes utilisant l'écrasement comme base de la reconstruction [ZPVG01, CDK04]. Leur idée est de projeter un quadrilatère portant une "empreinte" de la fonction noyau. De nombreuses fonctions noyau différentes ont été proposées, la plus efficace semblant être la projection d'un disque (ellipse).

Reprenant le principe de l'écrasement de photons, nous avons choisi de représenter le photon par une petite sphère. Le rayon de cette sphère est la distance maximale à laquelle un rayon peut-être influencé. Cela correspond en fait à la taille du support de l'estimation par noyau. A l'intérieur de cette sphère, la répartition de l'énergie est défini par la fonction noyau 3D choisie pour la reconstruction. Tout comme pour le rendu volumique, il est plus rapide de projeter un disque qu'une sphère. Nous souhaitons donc faire de même. Or, comme le montre la figure 5.7, la contribution du photon pour chaque rayon qui traverse sa zone d'influence dépend de la distance traversée et du filtre choisi.

Choix du filtre de reconstruction Le poids dû à la fonction noyau utilisée doit être calculé pour chaque rayon issu des pixels recouverts par la projection du photon, et ce pour chaque photon. Or le calcul est très similaire. Nous précalculons donc une texture de reconstruction

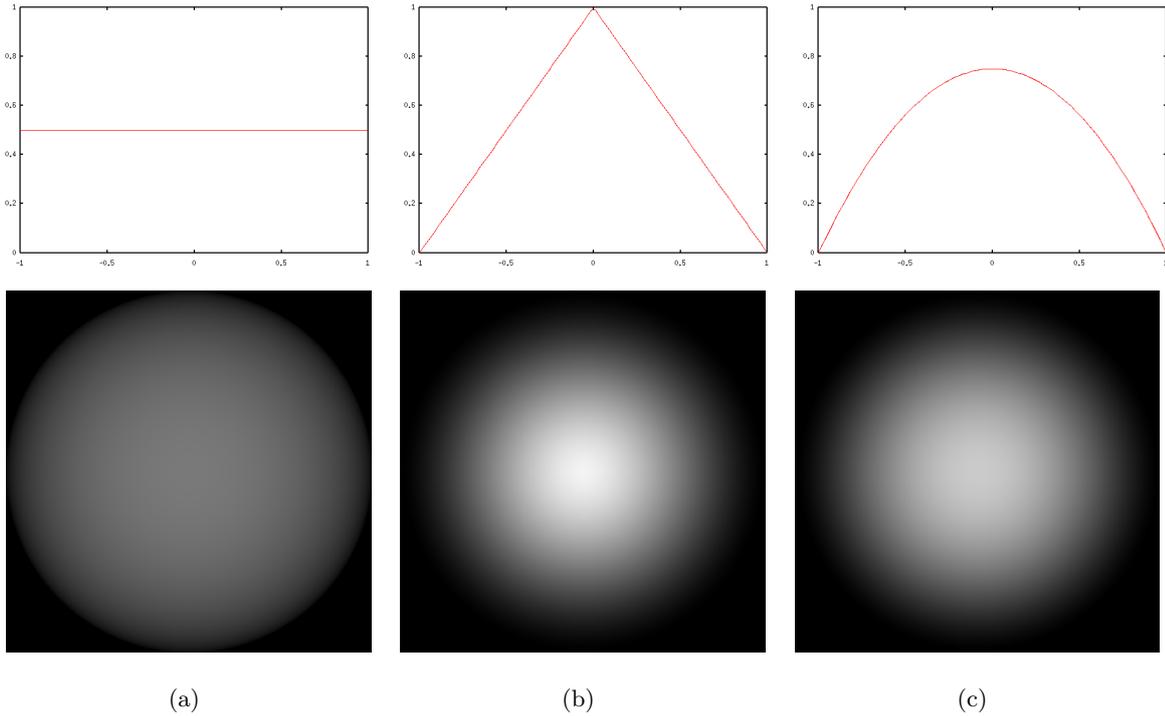


Figure 5.8. Profil 1D des différentes fonctions noyaux et les textures intégrées associées.
 (a) Filtre constant. (b) Filtre conique. (c) Filtre d'Epanechnikov.

représentant la fonction noyau intégrée sur le trajet parcouru par le rayon en fonction de sa distance au centre de la sphère. Nous choisissons donc de représenter le photon comme un quadrilatère toujours orienté vers l'observateur (i.e. *billboard*) portant cette texture. Nous faisons alors l'approximation que le poids associé au rayon issu d'un pixel recouvert par la projection du billboard est égal à la valeur de la texture en ce point. Cela introduit une petite erreur, mais permet d'accélérer le calcul de manière importante car il n'est plus nécessaire de calculer l'intégrale pour chaque rayon.

La valeur de chaque pixel (u, v) de la texture est égal à :

$$T(u, v) = \frac{1}{\frac{4}{3}\pi r^3} \int_A^B K(x_{u,v}(t) - x_p) dt \quad (5.17)$$

où r est le rayon du photon virtuel et A et B sont les point d'entrée et de sortie du rayon dans le photon comme sur la figure 5.7.

Le choix du filtre influence grandement la convergence de la méthode. On peut voir sur la figure 5.8, les 3 filtres que nous avons testés. Dans l'ordre, il s'agit du filtre constant, du filtre conique et enfin du filtre d'Epanechnikov qui est connu en statistiques pour être le filtre optimal.

Filtre constant Ce filtre répartit également l'énergie du photon sur tout le volume de la sphère. Il est le filtre le plus rapide à évaluer puisqu'il ne nécessite pas le calcul de la distance au centre de la sphère. Néanmoins, la convergence de la méthode est réduite car il génère des artefacts francs (i.e. ajout de disques aux bords nets les uns avec les autres) dans les images produites. Il n'est intéressant que lorsque le nombre de photons est très important.

$$F(x, y, z) = \begin{cases} \frac{1}{\frac{4}{3}\pi r^3} & \text{si } \sqrt{x^2 + y^2 + z^2} \leq r, \\ 0 & \text{sinon.} \end{cases}$$

Filtre conique Le filtre conique correspond à une variation linéaire de l'énergie à mesure que l'on s'éloigne du centre de la sphère. Il augmente grandement la convergence de l'algorithme car il génère moins d'artefact. Cependant, son calcul nécessite le calcul de la distance au centre, i.e. une racine carrée. Ce qui est relativement coûteux.

$$F(x, y, z) = \begin{cases} \frac{1}{\frac{4}{3}\pi r^3} \left(1 - \frac{\sqrt{x^2 + y^2 + z^2}}{r}\right) & \text{si } \sqrt{x^2 + y^2 + z^2} \leq r, \\ 0 & \text{sinon.} \end{cases}$$

Filtre optimal Le filtre d'Epaneshnikov est celui qui propose la convergence la plus rapide. Il a été prouvé [Sil86] que si l'on a besoin de N photons pour faire un rendu avec un filtre constant, $0.93N$ photons seulement sont nécessaires pour obtenir la même précision en utilisant ce filtre. De plus, l'évaluation de sa valeur est plus rapide que pour le filtre conique (pas de racine carrée).

$$F(x, y, z) = \begin{cases} \frac{1}{\frac{4}{3}\pi r^3} \left(1 - \frac{|x^2 + y^2 + z^2|}{r^2}\right) & \text{si } \sqrt{x^2 + y^2 + z^2} \leq r, \\ 0 & \text{sinon.} \end{cases}$$

Atténuation du photon Pour chaque photon, on calcule l'atténuation $P(j_p)$ de sa puissance par le milieu. Pour cela, on lance un rayon depuis l'oeil vers le photon et on applique la méthode du ray marching sur ce rayon. Nous faisons ici l'approximation que l'atténuation correspondant à chacun des pixels couverts par la projection du rayon est égale à l'atténuation du photon. Cette approximation introduit du biais dans la solution. Néanmoins, ce biais est proportionnel à la largeur du support h de l'estimation de densité, et la simulation converge vers la solution réelle quand h tend vers 0. Comme nous traçons un rayon par photon pour calculer l'atténuation, nous faisons des calculs redondants que la méthode de Jensen ne fait pas. Mais, comme illustré dans la section résultats 5.8.4, notre méthode reste plus rapide que la méthode des cartes de photons classique.

Fonction de phase La puissance du photon est modulée par la valeur de la fonction de phase entre la direction du photon et la direction de l'oeil.

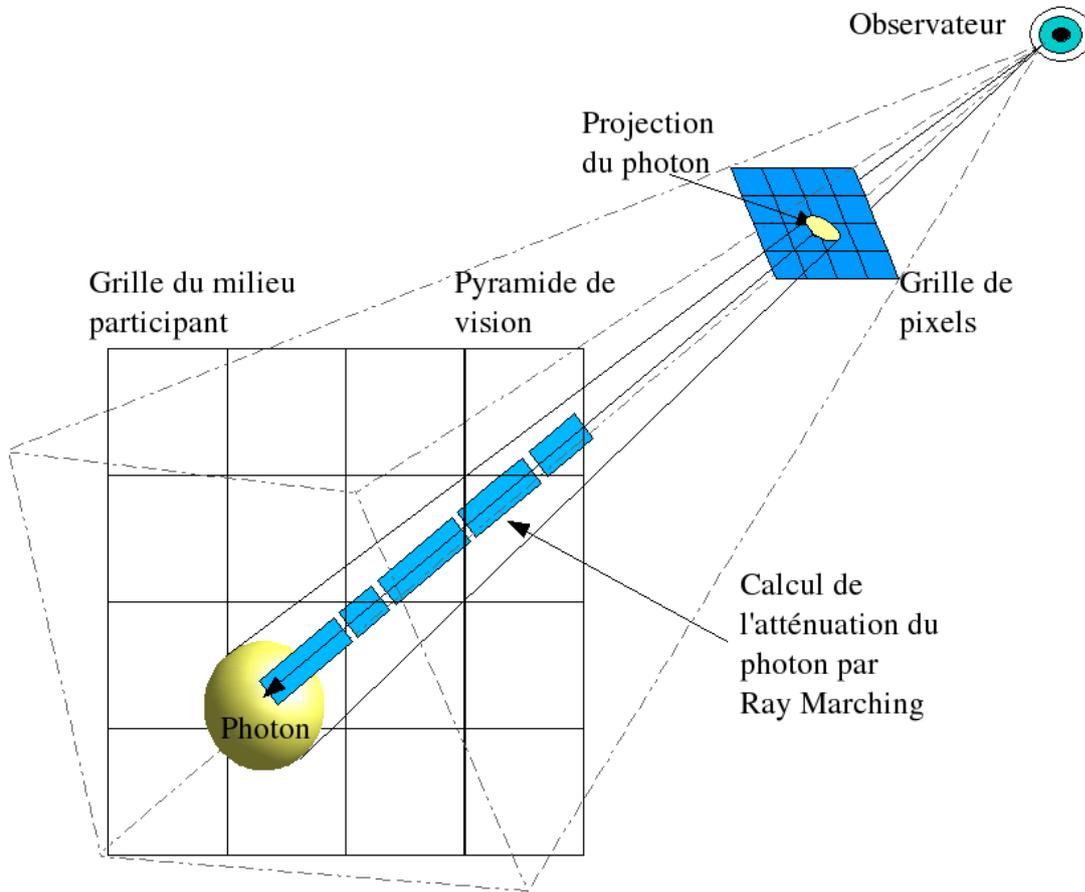


Figure 5.9. Principe de l'écrasement de photons volumique. La sphère associée au photon est projetée sur l'écran. L'atténuation du photon est calculée par Ray Marching. Chaque pixel recouvert par la projection est susceptible de recevoir une contribution non nulle.

Récapitulation Au final, pour chaque photon, on doit projeter un quadrilatère texturé sur l'écran, calculer l'atténuation, et pour chaque pixel recouvert, on ajoute la contribution du photon, à savoir, la puissance du photon modulée par les poids de la texture et de la fonction de phase ainsi que par l'atténuation.

5.7 Algorithme de l'écrasement de photons

Une fois que nous avons calculé d'une part l'émission, l'absorption et la diffusion sortante, et d'autre part la diffusion entrante, la dernière chose à faire est de sommer les deux images. L'algorithme 4 résume notre méthode.

Sélectionner les photons visibles

Pour chaque photon visible avec la puissance Φ_p **faire**

Calculer le poids w du photon par Ray Marching

Projeter le photon sur l'écran

Pour chaque pixel p couvert par la projection du photon **faire**

Récupérer le poids w_f depuis la texture du filtre d'estimation

$$L(p)+ = w * w_f * \Phi_p$$

Fin pour

Fin pour

Calculer l'image de l'émission, de l'absorption et de la diffusion sortante par Ray Marching

Sommer l'image précédente avec l'image de diffusion entrante.

Algorithme 4. *Algorithme de l'écrasement de photons.*

5.8 Optimisation GPU

Ces dernières années, l'optimisation de nombreux types de calculs par l'utilisation des processeurs graphiques (GPU i.e. *Graphic Processor Unit*) a connu un essor très rapide. En effet, ces processeurs effectuent du calcul parallèle optimisé. Auparavant, ces capacités de calcul étaient très difficiles à exploiter. En effet, la chaîne de traitement graphique était vue comme une boîte noire du point de vue de l'utilisateur. Cependant, avec l'apparition des unités programmables (i.e. *shaders*), tout d'abord accessible grâce à un code assembleur, puis par des langages de haut niveau de type C (e.g. GLSL, Cg), les possibilités se sont multipliées. Avec le rajout de la dernière brique manquante, à savoir l'utilisation de textures flottantes tant en entrée qu'en sortie du processeur graphique, il est devenu possible d'effectuer tous les calculs habituellement effectués sur le CPU.

Notre méthode utilise le principe de la projection des photons sur l'écran. Or les cartes graphiques sont particulièrement adaptées à ce type de traitement, les textures flottantes nous

permettant aujourd’hui d’obtenir une précision satisfaisante. Nous allons décrire la façon dont nous avons adapté la méthode de l’écrasement de photons au calcul par GPU.

5.8.1 Notions d’implémentation GPU

La programmation des cartes graphiques intervient sur deux étages distincts de la chaîne de traitement des cartes graphiques (cf. figures 5.10 et 5.11). Il s’agit de l’étage de traitement des sommets (i.e. *Vertex Shader*) et des pixels de l’image (i.e. *Fragment Shader*). Lorsque la programmation est utilisée, les étages de transformation et d’éclairage sont remplacés par l’exécution du code sur les unités dédiées au Vertex Shader et l’étape de multi-texturation est remplacée par l’exécution du code sur celles dédiées au Fragment Shader. On peut alors affecter n’importe quel traitement aux données sommets, puis les résultats obtenus sont interpolés pour tous les fragments de l’image (rasterisation) qui peuvent subir à leur tour un second traitement.

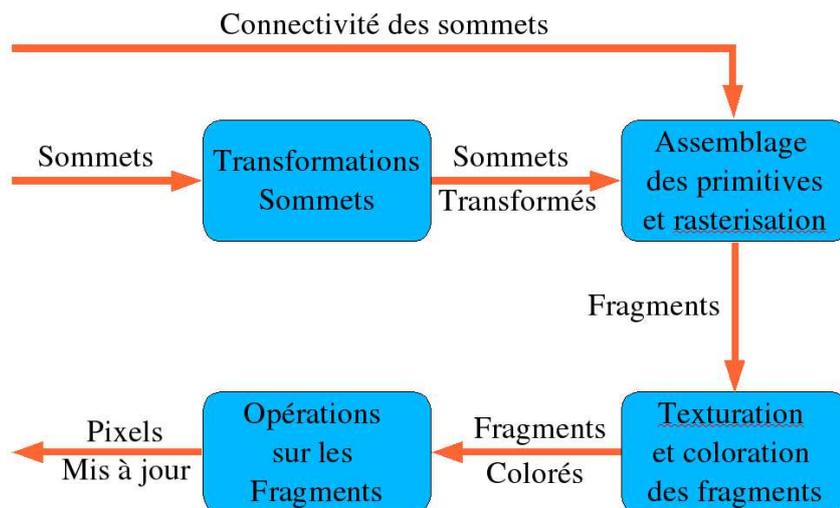


Figure 5.10. Pipeline OpenGL classique.

5.8.2 Principe

Notre idée est de tirer avantage de la projection optimisée et des capacités de texturations des processeurs graphiques afin de réduire de beaucoup le temps de rendu. En effet, ce sont les points de notre algorithme qui sont les plus gourmands en temps de calcul.

Nous nous arrangeons pour profiter au maximum du parallélisme entre le processeur central et le processeur graphique. Nous nous positionnons dans le mode de rendu dans un tampon

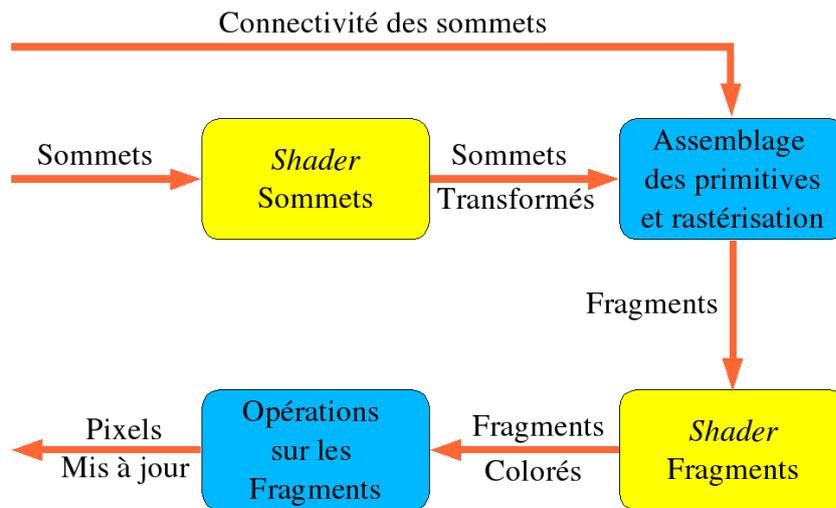


Figure 5.11. Pipeline OpenGL programmable.

flottant hors écran. La texture du filtre choisi est générée et chargée en mémoire graphique. Il s'agit d'une texture flottante avec 16 bits de précision. Pour chaque photon, l'atténuation et le poids de la fonction de phase sont calculés par le processeur central de la même manière que dans la section précédente. On calcule alors la puissance finale du photon en multipliant sa puissance par les valeurs précédemment calculées. Alors, le quadrilatère représentant le photon est envoyé au processeur graphique en associant aux sommets leurs coordonnées pour l'accès à la texture filtre (premières coordonnées de texture) et la puissance finale en tant que coordonnées de texture secondaires. Nous devons effectuer cela car les coordonnées de couleurs sont tronquées lorsqu'elles sortent des bornes $[0, 1]$. Nous utilisons le Pixel Shader uniquement pour moduler la puissance par la texture de filtre. La luminance est accumulée dans le tampon de rendu en utilisant les capacités de mélange du GPU pour les flottants 16 bits .

La génération de l'image d'absorption, émission et diffusion sortante n'est pas modifiée et est toujours effectuée sur le CPU. L'optimisation GPU ne porte que sur le calcul de la diffusion sortante. Une fois ce calcul effectué, on lit le tampon de rendu et on l'ajoute à l'image finale.

5.8.3 Précision

Problème Malheureusement, les processeurs graphiques actuels ne permettent d'effectuer du mélange flottant qu'avec une précision de 16 bits. Ce qui provoque des problèmes de précision. En effet, nous devons projeter et mélanger la luminance de tous les photons dans les pixels de l'écran. Pour un format flottant codé sur 16 bits, on perd de la précision lorsque l'on ajoute des petits nombres avec des grands. Or, nous devons mélanger de petites énergies de photons

un grand nombre de fois par pixel, et après un certain nombre d'additions (i.e. un millier de photons par pixel), la valeur du pixel devient grande devant le poids du photon, ce qui cause une perte de précision. Néanmoins, cette perte de précision n'est un problème que lorsque les images générées sont utilisées pour effectuer des mesures de simulation physique, mais ce n'est pas un problème si l'on veut effectuer du rendu pseudo-réaliste ayant pour but la communication ou le loisir.

Résolution Ce problème de précision sera résolu naturellement si les processeurs graphiques viennent à permettre le mélange flottant 32 bits. En attendant, nous avons résolu le problème en utilisant une méthode multi-passes. Afin d'éviter la perte de précision, nous ne traitons dans chaque passe qu'un nombre de photons qui assure de ne pas perdre de précision. Nous lisons entre chaque passe le tampon de rendu et l'ajoutons à un tampon flottant en utilisant le processeur central. Cela cause évidemment une perte de performances, due à la lecture du tampon de rendu depuis la carte graphique vers la mémoire centrale. Nous pensons néanmoins que notre méthode donne encore de bons résultats. De plus, avec la sortie du port PCI express, la communication de la carte graphique vers la mémoire centrale devient aussi rapide que l'inverse. Ce qui réduit encore l'inconvénient de cette approche multi-passes.

5.8.4 Résultats



Figure 5.12. Rendu d'un nuage anisotropique en utilisant une carte de quatre millions de photons. 5.12(a) : Cartes de photons. 5.12(b) Notre algorithme.

	Luminance moyenne	Ecart type	Temps de rendu (s)
Cartes de photons	29.55	34.37	99096
Notre méthode	30.82	35.96	1497

Table 5.1. Comparaison de la luminance moyenne des pixels et écart type pour les images de la figure 5.12.

Afin de mesurer l'efficacité de notre méthode, nous avons fait des tests de performances. Ces tests ont été effectués sur un Pentium 4 cadencé à 2.4GHz muni qu'une Geforce 6800 GT de chez NVIDIA. Ils consistent à rendre un nuage anisotropique de $2000 \times 2000 \times 2000$ mètres. Il est modélisé par une grille de $60 \times 60 \times 60$ voxels. La figure 5.12 montre les images obtenues avec la méthode des cartes de photons et notre méthode d'écrasement de photons. La table 5.1 donne une comparaison de la qualité des résultats obtenus.

Les tables 5.2, 5.3 et 5.4 montrent respectivement les temps de calcul en fonction de la taille de l'image, du nombre de photons stockés et de la taille des photons utilisée. Les temps de calcul de notre méthode dans sa version normale (CPU) et dans sa version optimisée (GPU) sont comparés avec ceux obtenus avec la méthode des cartes de photons. La configuration des paramètres ne variant pas est :

- une image de 512×512 pixels,
- une carte de un million de photons,
- une taille de 10.0 mètres pour le photon.

La figure 5.13 montre une image de parabole spéculaire placée dans un milieu participant homogène éclairé par une source étendue placée loin devant la parabole.

Grâce à ces résultats, nous pouvons calculer le rapport d'accélération de notre méthode par rapport à la méthode des cartes de photons. Notre méthode est plus de dix fois plus rapide que celle de Jensen dans sa version logicielle. Néanmoins, elle a une complexité quadratique dépendant de la taille de l'image et la taille du photon, et une complexité linéaire sur la taille de la carte de photons. Cependant, elle reste plus rapide que la méthode des cartes de photons dans tous les cas d'utilisation classiques. Les temps de calculs que nous obtenons, bien que plus rapides, sont encore trop importants. Ceci est dû à notre implémentation logicielle de l'algorithme qui est dépendante de la résolution de l'image et de la taille du photon. En effet, tous les pixels couverts par la projection du photon doivent être traités l'un après l'autre de

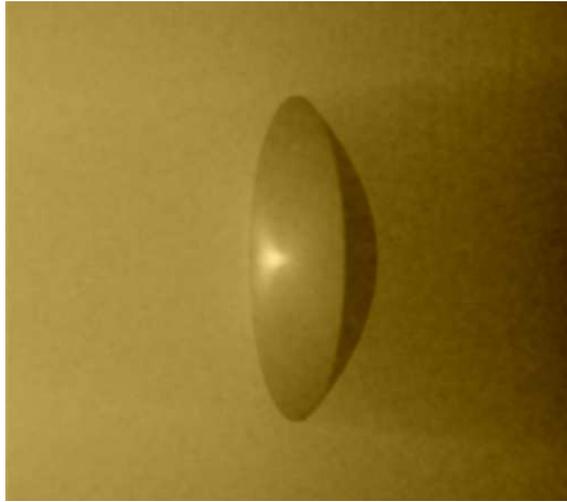


Figure 5.13. Rendu d'une parabole placée dans nuage homogène éclairé par une source étendue distante. La parabole génère une caustique et une ombre volumique.

manière à calculer la contribution du photon à chacun de ces pixels. Pour chaque pixel, nous devons accéder à la texture du photon, à l'énergie du photon, calculer le poids de la fonction de phase et en faire le produit. Ces opérations sont peu coûteuses individuellement mais deviennent problématiques lorsqu'elles doivent être effectuées de nombreuses fois. C'est pourquoi le temps de calcul augmente avec la résolution de l'image et la taille du photon utilisée. Ce problème disparaît quasiment avec la version optimisée par carte graphique. En effet, la projection des photons est alors optimisée par le matériel graphique levant ainsi la dépendance sur la taille de l'image est du photon. On obtient alors un ratio d'accélération de 30 dans le pire cas et 811 lorsque l'image et la taille du photon deviennent grandes. La seule dépendance restante concerne la taille de la carte de photon.

5.9 Conclusion et perspectives

Nous avons présenté la méthode de l'écrasement de photons volumique qui réduit considérablement les temps de calcul pour l'illumination globale dans les milieux participants. Nous avons reformulé l'équation volumique du transfert radiatif en une expression itérative, ce qui nous a permis d'extraire le calcul de la diffusion sortante du Ray Marching. Nous pouvons ainsi calculer séparément l'émission, l'absorption et la diffusion sortante d'un côté et la diffusion entrante de l'autre. Notre méthode optimise le temps de rendu, spécialement pour la diffusion entrante qui constitue la partie coûteuse du calcul, en utilisant une approche duale de l'estimation de densité.

Taille de l'image	256×256	512×512	1024×1024	2048×2048
Cartes de photons	912	4431	16005	63514
Notre méthode (CPU)	64	196	741	3463
Rapport d'accélération (CPU)	14	22	21	18
Notre méthode (GPU)	23.34	23.79	34.71	39.59
Rapport d'accélération (GPU)	39	186	461	1601

Table 5.2. Comparaison des temps de rendu (en secondes) obtenus avec les cartes de photons et notre méthode dans sa version logicielle ainsi que dans sa version optimisée par GPU en fonction de la taille de l'image. Ces résultats sont donnés pour une carte d'un million de photons et une taille de photon (i.e. distance de recherche pour les cartes de photons) de dix mètres. Le nuage utilisé est le même que dans la figure 5.12.

Nous avons montré que notre méthode était, dans sa version CPU, dix fois plus rapide que la méthode des cartes de photons dans le pire cas, pour une qualité équivalente. Nous proposons également une optimisation utilisant les processeurs graphiques qui permet de diviser à nouveau les temps de rendu par trois.

Nous utilisons actuellement une estimation de densité par noyau avec un paramètre de lissage constant, ce qui peut causer le lissage de certains détails. Nous pensons à utiliser l'estimation de densité adaptative de manière à adapter la taille du filtre en fonction des propriétés du milieu et la densité de photons. Cela serait adapté à l'amélioration des frontières nettes du milieu ainsi qu'à l'accélération de la convergence des caustiques, rendant ainsi les nuages générés encore plus réalistes.

Nous avons proposé l'utilisation de plusieurs filtres de reconstruction, néanmoins, il doit être possible d'accélérer encore le temps de rendu en appliquant strictement les méthodes de rendu volumique proposées dans [ZPVG01, CDK04] dont nous nous sommes inspirés. Enfin, nous calculons l'atténuation pour chaque photon en utilisant un ray marching. Nous pensons mettre en place une stratégie de projection de l'avant vers l'arrière de façon à calculer l'atténuation de manière incrémentale et éviter ainsi des calculs redondants.

Nombre de photons	1M	2M	3M	7M
Cartes de photons	4431	7424	9629	14781
Notre méthode (CPU)	196	381	514	1191
Rapport d'accélération (CPU)	22	19	18	12
Notre méthode (GPU)	23.79	48.57	72.92	167.71
Rapport d'accélération (GPU)	186	152	132	88

Table 5.3. Comparaison des temps de rendu (en secondes) obtenus avec les cartes de photons et notre méthode dans sa version logicielle ainsi que dans sa version optimisée par GPU en fonction de la taille de la carte de photons. Ces résultats sont donnés pour une image de 512×512 pixels et une taille de photon (i.e. distance de recherche pour les cartes de photons) de dix mètres. Le nuage utilisé est le même que dans la figure 5.12.

Taille du photon	1.0	5.0	10.0	20.0	30.0
Cartes de photons	749	1322	4431	13649	21913
Notre méthode (CPU)	32	67	196	708	1563
Rapport d'accélération (CPU)	23	19	22	19	14
Notre méthode (GPU)	23.92	24.09	23.79	23.79	27.01
Rapport d'accélération (GPU)	31	54	186	573	811

Table 5.4. Comparaison des temps de rendu (en secondes) obtenus avec les cartes de photons et notre méthode dans sa version logicielle ainsi que dans sa version optimisée par GPU en fonction de la taille de la carte de photons. Ces résultats sont donnés pour une image de 512×512 pixels et une carte d'un million de photons. Le nuage utilisé est le même que dans la figure 5.12.

Conclusion

Au cours de cette thèse, nous nous sommes penchés sur l'amélioration de la simulation optronique qui regroupe les domaines du visible et de l'infrarouge, et plus particulièrement sur le calcul des phénomènes de diffusion. Nous avons proposé deux méthodes : le lancer de photons multi-passes et l'écrasement de photon volumique, la première ayant pour but de résoudre les problèmes de mémoires inhérents au traitement de scènes complexes, la seconde l'optimisation des temps de calcul pour le rendu de milieux participants (fumées, brouillards, gaz chauds...). Ces méthodes ont été industrialisées dans le logiciel SPECRAY de la société OKTAL Synthetic Environment (cf annexe C.1), ce qui a permis leur validation. Nous avons également participé à la conception et à l'implémentation de la nouvelle version du logiciel de visualisation infrarouge temps-réel COMPIR (annexe C.2). Enfin Nous avons conçu un module fournissant un modèle d'éclairage et de ciel pour augmenter le réalisme des simulateurs temps réel de la société OKTAL (annexe C.3.1).

Grâce à un tour d'horizon complet des méthodes existantes de synthèse d'images, nous avons montré que la méthode des cartes de photons était la plus susceptible d'être adaptée au rendu spectral infrarouge. D'une part, elle propose une méthode unifiée de traitement des surfaces et des milieux participants. D'autre part, il s'agit de la méthode offrant le meilleur compromis entre qualité de simulation et temps de rendu. Comme la plupart des autres méthodes de synthèse d'images, elle a été proposée pour la simulation du domaine visible uniquement.

Notre première contribution a été d'étendre la méthode au domaine infrarouge. Nous avons proposé une évolution spectrale de la méthode qui permet de s'adapter à n'importe quelle découpe spectrale de l'infrarouge et du visible. Néanmoins, celle-ci souffrait du même problème de consommation mémoire que la méthode originale. En effet, la précision de la simulation augmente avec le nombre de photons stockés. Pour obtenir une précision suffisante pour des matériaux complexes, il est nécessaire de stocker un très grand nombre de photons, ce qui est

irréalisable à moins d'avoir une architecture matérielle très onéreuse. En reformulant le problème de l'illumination globale, nous avons mis au point une méthode fonctionnant sur un principe d'estimation de densité multi-passes. Nous avons en effet montré l'équivalence entre l'estimation classique de la luminance à partir de la carte de photons et la moyenne des estimations effectuées sur des petites cartes formant une partition de la carte originelle. Cela donne l'avantage à notre méthode de pouvoir fonctionner sur la plupart des ordinateurs, et d'obtenir malgré tout la précision souhaitée. La convergence de notre méthode fournit rapidement une bonne approximation de la solution, puis, au prix de suffisamment de temps de calcul, permet d'obtenir une grande précision. Ces temps de calcul peuvent être accélérés en utilisant une ferme de processeurs, notre méthode étant particulièrement adaptée au parallélisme.

Les performances de la méthode pourraient encore être améliorées en utilisant un modèle spectral plus efficace. De plus, la mise en place d'un système d'analyse des passes de calcul précédentes permettrait de concentrer l'effort de calcul en priorité sur les zones où la précision est nécessaire, améliorant ainsi la convergence de la méthode.

Notre deuxième contribution concernait la simulation efficace des milieux participants. La méthode des cartes de photons prend en compte ce type de milieu. Néanmoins ses performances sont déplorables. Nous avons proposé une méthode innovante, conjuguant rendu volumique et estimation de densité, qui minimise les temps de calcul pour le rendu de tels milieux. Ce nouvel algorithme est dix fois plus rapide que les cartes de photons dans le pire des cas tout en conservant une qualité identique. De plus, il est particulièrement adapté à l'optimisation par la programmation du matériel graphique, ce qui nous permet d'obtenir finalement des temps de calcul trente fois inférieurs à la méthode des cartes de photons, toujours dans le pire des cas, notre méthode pouvant être jusqu'à 800 fois plus rapide.

Les résultats obtenus par notre méthode d'écrasement de photons volumiques sont très encourageants dans le cadre de la simulation hors temps réel. Il ne manque pas grand chose pour pouvoir effectuer du rendu interactif voire temps réel. Bien que nous nous soyons inspirés du rendu volumique, nous n'avons pas mis en place de manière rigoureuse les optimisations que ces méthodes proposent (par manque de temps). Il serait, par exemple, judicieux d'étudier l'impact du choix du filtre sur la rapidité de la simulation, ainsi qu'une stratégie de projection des photons de l'avant vers l'arrière afin d'optimiser les calculs d'atténuation.

Annexe A

Données radiatives

A.1 Flux

Le flux représente l'énergie arrivant ou quittant une surface par unité de temps. Il est noté Φ et s'exprime en Watt (W).

A.2 Eclairement

L'éclairement est le flux par unité de surface. Il est noté E et s'exprime en Watt par mètre carré ($W.m^{-2}$).

A.3 Luminance

La luminance est le flux par unité de surface projetée et par unité d'angle solide. Elle est notée L et s'exprime en Watt par stéradian par mètre carré ($W.sr^{-1}.m^{-2}$).

A.4 Fonction de réflectance bidirectionnelle

La fonction de réflectance bidirectionnelle (FDRB) caractérise les propriétés de réflexion d'une surface au niveau macroscopique. Intuitivement, elle permet de connaître l'énergie quittant une surface en fonction de l'énergie incidente (et vice versa).

La définition formelle de la FDRB a été introduite par Nicodemus [NRH⁺77]. Etant données une direction incidente ω_i et une direction de réflexion ω_r , la FDRB est le rapport entre la luminance réfléchie en un point x associé à une surface élémentaire d'aire dA et l'éclairement

reçu par celle-ci. La FDRB est généralement exprimée dans le repère local à la surface, repère défini par la normale à cette surface. Pour une longueur d'onde λ , on a :

$$f_r(x, \omega_i, \omega_r, \lambda) = \frac{dL_r(x, \omega_r, \lambda)}{dE_i(x, \omega_r, \lambda)} = \frac{dL_r(x, \omega_r, \lambda)}{L_i(x, \omega_i, \lambda) \cos \theta_i d\omega_i} \quad (\text{A.1})$$

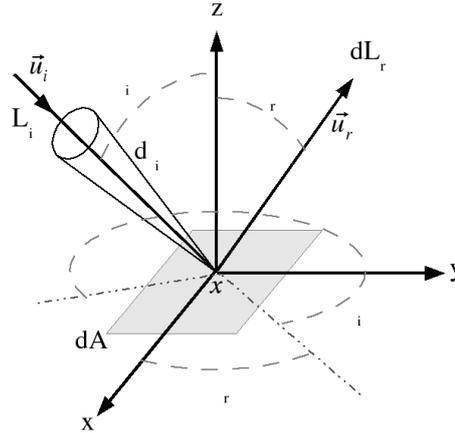


Figure A.1. Géométrie locale à la surface associée à la formulation de la FDRB.

Aujourd'hui il existe des appareils de mesure, les goniomètres, qui permettent de mesurer la FDRB des matériaux. Cependant, les mesures issues de ces appareils sont difficilement exploitables en l'état. D'une part, les données obtenues sont souvent bruitées et incomplètes, et nécessitent donc d'être traitées avant utilisation. D'autre part, le coût de stockage nécessaire est prohibitif pour l'utilisation. En effet, les données sont mesurées pour un certain nombre de directions, et la précision de la mesure augmente avec ce nombre. Pour avoir des mesures précises il est donc nécessaire de stocker un grand volume de données. On doit donc soit se limiter à des modèles simples de BRDF, soit mettre en oeuvre des méthodes de compressions efficaces. Claustres [Cla03] propose par exemple un modèle de compression par ondelettes sphériques.

A.5 Fonction de phase

Une fonction de phase est une fonction qui, étant donnée une direction, donne la quantité d'énergie lumineuse redirigée dans chacune des directions de l'espace lorsqu'il y a interaction avec une particule du milieu participant. Elle joue un rôle semblable à celui de la BRDF pour les surfaces. On note $p(x, \omega, \omega')$ la fonction de phase au point x , pour la direction d'incidence ω et la direction sortante ω' . La fonction de phase possède la propriété de conservation de l'énergie :

$$\int_{\Omega=4\pi} p(x, \omega, \omega') d\omega' = 1 \quad (\text{A.2})$$

La fonction de phase ne dépend que de l'angle θ entre la direction incidente et la direction de diffusion du rayonnement.

A.6 Loi du corps noir

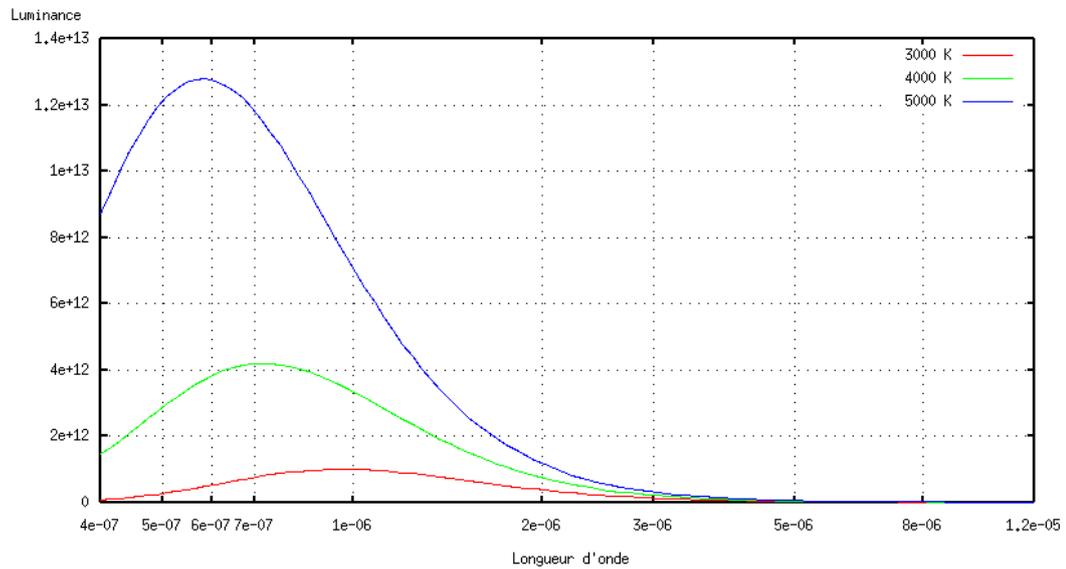


Figure A.2. Luminance du corps noir donnée par la loi de Planck pour des températures de 3000, 4000 et 5000 Kelvins.

La luminance spectrale émise par un corps noir (corps totalement absorbant et émissif) à la température T est donnée par la loi de Planck, ou loi du corps noir (cf. figure A.2) :

$$\frac{dL(\lambda, T)}{d\lambda} = \frac{2hc^2\lambda^{-5}}{\exp\left(\frac{hc}{\lambda kT}\right) - 1}$$

où :

- $\frac{dL(\lambda, T)}{d\lambda}$ est la quantité d'énergie émise par unité de temps, par unité de surface, par unité d'angle solide, par unité de longueur d'onde (e.g. $W.m^{-2}.sr^{-1}.m^{-1}$),
- λ est la longueur d'onde (m^{-1}),
- T est la température du corps noir (K),
- $h = 6.6260693 \times 10^{-34} J.s$ est la constante de Planck

- $c = 299792458 m.s^{-1}$ est la vitesse de la lumière dans le vide
- $k = 1.3806505 \times 10^{-23} J/K$ est la constante de Boltzmann.

Max Planck proposa originellement cette loi in 1900 pour essayer d'améliorer une expression proposée par Wilhelm Wien qui respectait les données expérimentales pour les ondes courtes mais qui déviait pour les longues. Il trouva que la fonction ci-dessus correspondait très bien aux données pour toutes les longueurs d'ondes.

Annexe B

Fonction de densité de probabilité

B.1 Définition

Une fonction de densité de probabilité (PDF) est utilisée pour représenter une distribution de probabilité en termes d'intégrales. Cette fonction $f(x)$ est toujours positive sur son domaine de définition et on a :

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

Etant donné une distribution de probabilité possédant la densité $f(x)$, alors l'intervalle infinitésimal $[x, x + dx]$ a la probabilité $f(x)dx$. Plus simplement, une fonction de densité de probabilité peut être vue comme une version lissée d'un histogramme : si on mesure empiriquement les valeurs d'une variable aléatoire continue de manière répétitive et qu'on construit un histogramme qui décrit le nombre de d'occurrence de chaque valeur de la variable, alors cet histogramme ressemble à la fonction de densité de probabilité de cette variable. Formellement, une distribution de probabilité a la densité $f(x)$ si $f(x)$ est une fonction positive intégrable de $\Re \rightarrow \Re$ telle que la probabilité P de l'intervalle $[a, b]$ est donnée par :

$$P[a \leq X \leq b] = \int_a^b f(x)dx$$

quelques soient a et b . La figure B.1 présente un exemple de fonction de densité de probabilité gaussienne.

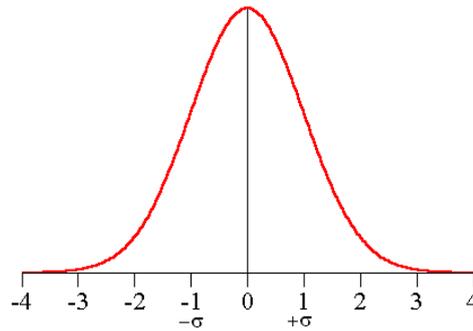


Figure B.1. Fonction de densité de probabilité gaussienne (ou loi normale).

B.2 Échantillonnage par importance

B.2.1 Principe

Les calculs de type intégration de Monte Carlo peuvent être effectués en utilisant un ensemble de points aléatoires tirés de n'importe quelle distribution de probabilité arbitraire. Le choix de la distribution influence grandement l'efficacité de la méthode. Dans la plupart des cas, l'utilisation de distributions de probabilités uniformes donne une estimation pauvre des intégrales sur un grand nombre de dimensions. Par conséquent, l'utilisation de ces distributions n'est pas une méthode d'approximation utile. En 1953, toutefois, Metropolis et son équipe ont introduit un nouvel algorithme d'échantillonnage de points utilisant une fonction de probabilité donnée. Cet algorithme permet l'incorporation de l'échantillonnage par importance dans l'intégration de Monte-Carlo. Au lieu de tirer des points avec une distribution uniforme, ceux-ci sont tirés avec une distribution qui concentre les points où la valeur de la fonction que l'on intègre est grande. L'intégrale peut alors s'écrire :

$$I = \int_a^b \frac{f(x)}{g(x)} g(x) dx$$

où la fonction $g(x)$ est choisie comme étant une approximation raisonnable de la fonction $f(x)$. L'intégrale peut alors être calculée en tirant aléatoirement les échantillons x_i avec la distribution de probabilité $g(x)$ en évaluant $\frac{f(x_i)}{g(x_i)}$. Pour que $g(x)$ puisse être utilisée comme fonction de densité, elle doit respecter la définition citée dans la section B.1, le meilleur choix possible étant $|f(x)|$. La valeur moyenne de ces évaluations donne une estimation de l'intégrale I .

B.2.2 Mise en oeuvre

Génération d'échantillons aléatoires pour une densité $p(x)$ dont la fonction de distribution cumulative est inversible Dans le cas où la fonction de distribution cumulative $P(x)$

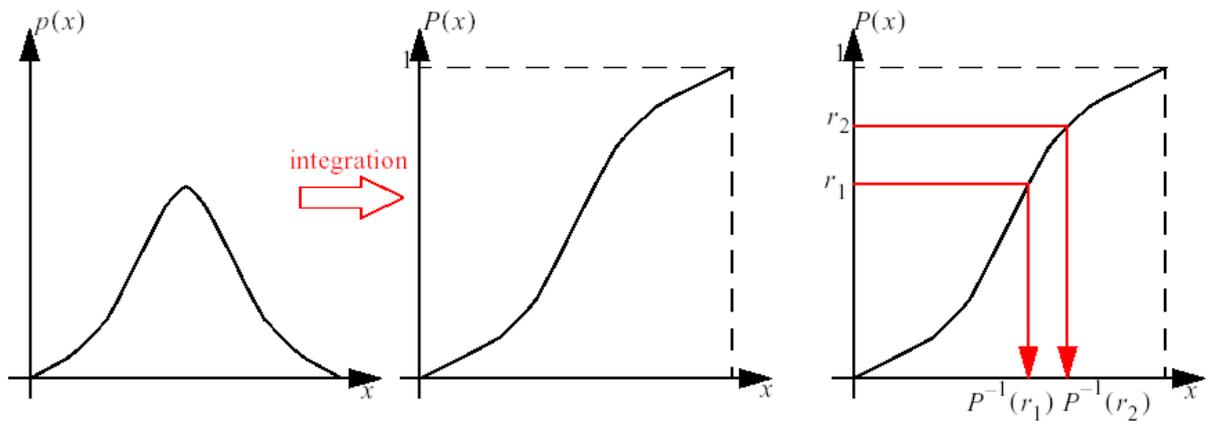


Figure B.2. Mécanisme d'échantillonnage selon une loi de probabilité inversible.

de la probabilité que l'on souhaite échantillonner est inversible, il suffit de calculer l'intégrale de la densité pour obtenir cette fonction cumulative, de tirer uniformément un nombre aléatoire r entre 0 et 1, puis de rechercher la valeur de la variable aléatoire x :

$$x = P^{-1}(r)$$

La figure B.2 présente le processus.

Génération d'échantillons par rejet Le tirage par importance présenté dans le paragraphe précédent ne fonctionne que si la fonction de distribution cumulative est inversible. Malheureusement, dans la plupart des cas elle ne l'est pas. On procède donc à un tirage par rejet. On définit un nombre M tel que : $\forall x : p(x) \leq M$. Ensuite, on génère un doublet de valeurs aléatoires $(x', y') \in D \times [0, M]$. Si $p(x') \leq y'$ on accepte l'échantillon x' , sinon on rejette l'échantillon et on refait un tirage. La figure B.3 présente le mécanisme.

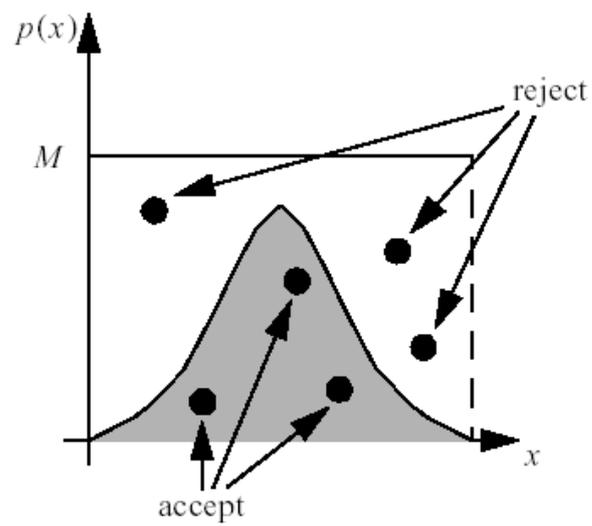


Figure B.3. Mécanisme d'échantillonnage par rejet.

Annexe C

Réalisations industrielles

C.1 SPECRAY

C.1.1 Description

SPECRAY est un logiciel de lancer de rayons développé au sein de la société OKTAL SE. Il permet le calcul d'images physiques spectrales d'un grand réalisme dans les spectres visible et infrarouge. SPECRAY est original car il s'appuie sur un modèle physique qui, à la différence de la plupart des lancer de rayons, permet le calcul des images, bande spectrale par bande spectrale. La figure C.1 présente des images générées dans différentes conditions d'observation.

C.1.2 Apports

Le premier apport que nous avons fourni à SPECRAY est la finalisation de l'implémentation de la méthode originale des cartes de photons afin d'ajouter la prise en compte des phénomènes de diffusion pour les surfaces. La méthode de lancer de photons multi-passes présentée au chapitre 4 a été mise au point afin de résoudre les problèmes de consommation mémoire excessive que connaissait SPECRAY pour le rendu de scènes infrarouges utilisant la diffusion. De la même façon, la méthode de l'écrasement de photons avait pour but d'améliorer la prise en compte des milieux participants diffusant. Le problème étant dans ce cas la longueur des temps de calcul. Nous avons implémenté les deux méthodes dans SPECRAY. Cette industrialisation a permis de tester et valider les algorithmes proposés dans cette thèse.

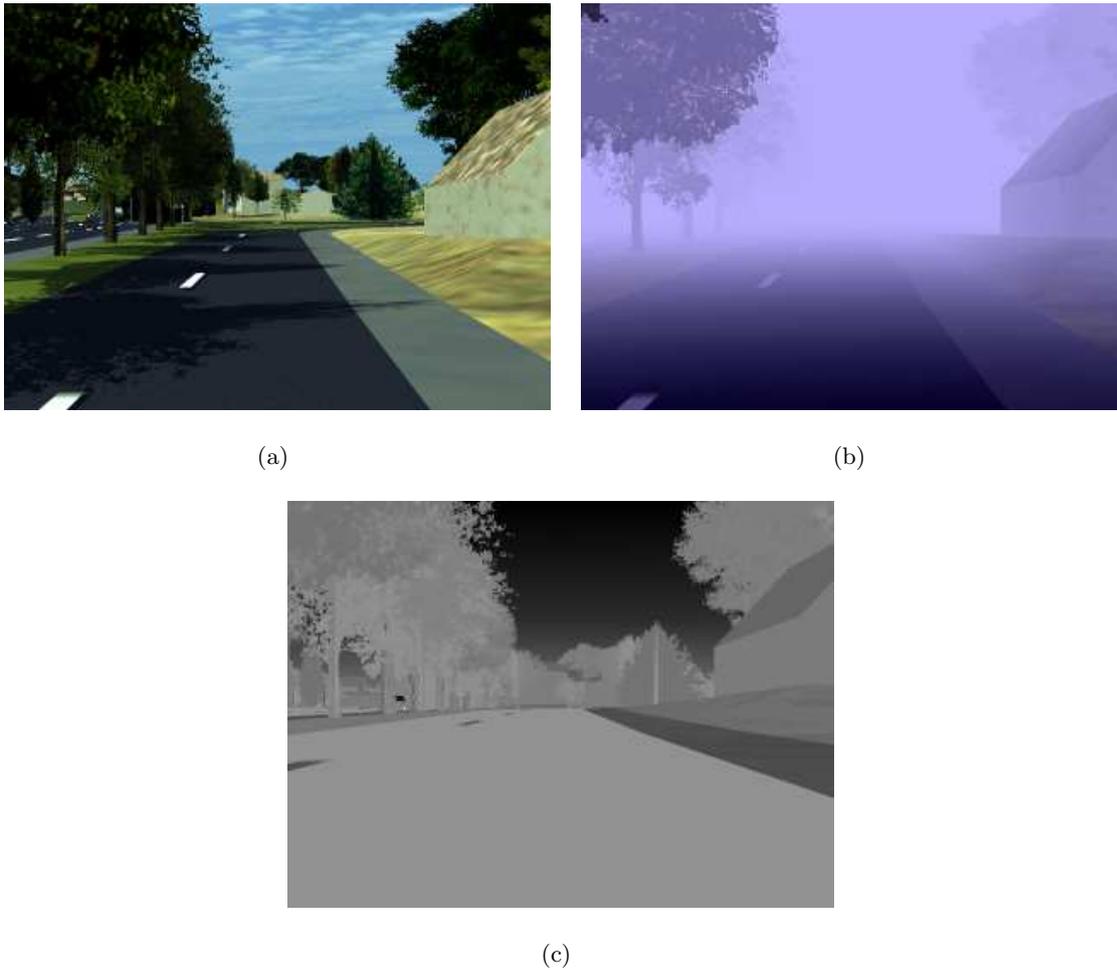


Figure C.1. Exemple d'images générées par SPECRAY pour un même point de vue dans différentes conditions d'observation. (a) Rendu visible par beau temps. (b) Rendu visible par temps de brouillard. (c) Rendu infrarouge bande III par beau temps.

C.2 COMPIR : Visualisation infrarouge temps réel

C.2.1 Description

COMPIR est l'équivalent temps réel de SPECRAY. Une compilation des données du modèle est effectuée afin de générer des textures de luminance qui sont affectées aux faces du modèle. La scène est ensuite visualisée grâce au rendu classique OpenGL. Le modèle est validé par la comparaison des résultats obtenus avec SPECRAY. Le logiciel permet par exemple de naviguer en temps réel sur des zones de terrain de grande taille aussi bien en visible qu'en infrarouge. La figure C.2 montre un exemple de visualisation d'un terrain de grandes dimensions dans les trois bandes infrarouges classiques.

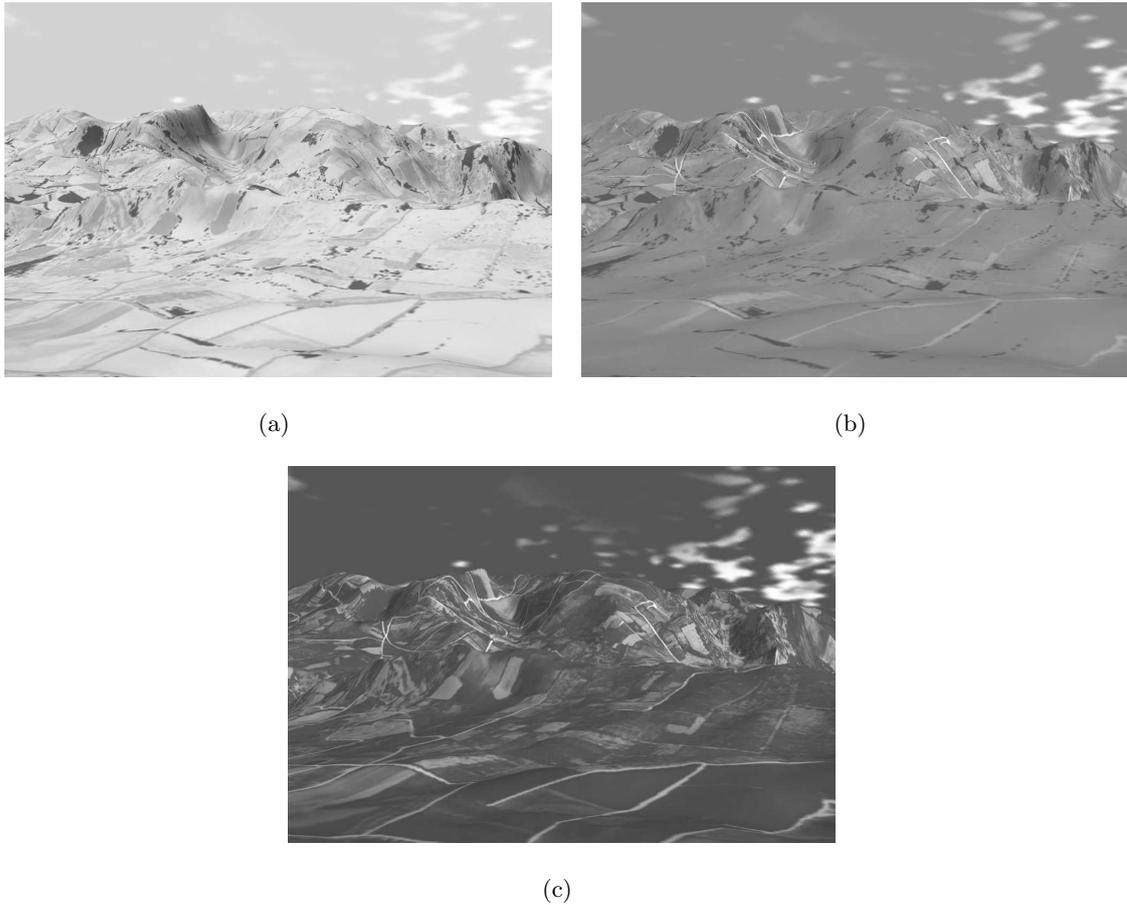


Figure C.2. Rendu temps réel d'un terrain de grandes dimensions portant une texture classifiée infrarouge générique par l'application COMPIR. (a) Bande I. (b) Bande II. (c) Bande III.

C.2.2 Apports

La compilation des données est coûteuse pour les scènes complexes. En effet, les calculs d'illumination mis en oeuvre pour la génération des textures de luminance sont très proches de ceux utilisés par SPECRAY. De plus, les bases de données 3D sont compilées pour une bande spectrale d'observation et un gain de luminance fixe. Ce qui nécessite de créer autant de bases de données que de conditions distinctes d'observation.

Nous avons participé à la conception d'une version améliorée de COMPIR utilisant la programmation des processeurs graphiques. On procède toujours à une compilation des bases de données d'entrée. Celle-ci consiste à stocker dans une texture flottante les différentes données radiatives des matériaux (émissivité, BRDF...). On stocke également dans une autre texture les températures des différents matériaux composant la scène en fonction de leur orientation. Ces

textures globales sont exploitées lors de la visualisation par les shaders de la carte graphique où s'effectuent tous les calculs d'illumination.

L'avantage de cette nouvelle approche est de pouvoir choisir la bande d'observation en temps réel ainsi que de régler le gain de luminance. On peut également faire évoluer les conditions atmosphériques, ou la température. On se débarrasse également du problème de stockage de plusieurs bases de données pour chaque condition d'observation qui sont problématiques en ce qui concerne le stockage disque.

C.3 RDVIS : Modèle de ciel temps réel

Le but du projet RDVIS était de proposer un ensemble de bibliothèques permettant d'uniformiser les différents simulateurs temps réels de la société OKTAL ainsi que les développements à venir. Ma participation dans le cadre de ce projet a été de concevoir le module d'éclairage avancé permettant d'augmenter le réalisme des simulations.

C.3.1 Modèles de ciel

Afin d'obtenir un éclairage de meilleure qualité sans modifier le modèle d'éclairage accéléré par les cartes graphiques, nous choisissons de représenter l'éclairage dû au ciel par une couleur ambiante pour la scène et une lumière directionnelle. Afin d'obtenir un éclairage réaliste, nous choisissons de caractériser ces valeurs à partir de modèles de ciels. En particulier, nous utilisons deux modèles de ciel donné par la Commission Internationale de l'Eclairage (CIE).

La figure C.3 présente une représentation 3D de différents modèles de ciel. Il s'agit de la luminance du ciel en fonction de la direction de l'élément de ciel.



Figure C.3. Représentation de modèles de ciel. De gauche à droite : ciel mitigé, très couvert et dégagé.

C.3.1.1 Modèle de ciel couvert

Pour les ciels couverts, la luminance au zénith L_z et la luminance $L(\theta)$ pour chaque élément de ciel (défini par θ , l'angle entre le zénith et l'élément de ciel) est donnée par les équations suivantes :

$$L_z = \frac{1}{0.203}(8.6\cos(\theta) + 0.123)$$

$$L(\theta) = L_z \frac{1}{3}(1 + 2\cos(\theta))$$

C.3.1.2 Modèle de ciel dégagé

Pour le calcul de la luminance au zénith L_z d'un ciel clair, un facteur de turbidité est défini.

$$L_z = \frac{1}{0.203}((1.367.turbidity - 1.81) \tan\left(\frac{\pi}{2} - \theta_s\right) + 0.38)$$

Pour chaque point (θ, ϕ) (θ définit l'angle de l'élément de ciel avec le zénith et ϕ est le gisement de l'élément de ciel) du dôme du ciel, la luminance est donnée par :

$$L(\theta, \phi) = L_z \frac{(0.91 + 10e^{-3\gamma} + 0.45 \cos^2 \gamma)(1 - e^{-\frac{0.32}{\cos \theta}})}{0.27385(0.91 + 10e^{-3\theta} + 0.45 \cos^2 \theta)}$$

où γ , l'angle calculé au centre du dôme entre le soleil et l'élément de ciel, est donné par :

$$\cos \gamma = \cos \theta_s \cos \theta + \sin \theta_s \sin \theta \cos(\phi - \phi_s)$$

où θ_s définit l'angle du soleil avec le zénith et ϕ_s est le gisement du soleil.

C.3.1.3 Interpolation entre modèles de ciel

Afin de pouvoir prendre en compte tous les types de ciel allant de couvert à dégagé, nous choisissons d'interpoler linéairement les valeurs calculées entre les deux modèles précédents. Le coefficient d'interpolation est défini par l'utilisateur. Il varie entre la valeur 0 pour un ciel couvert et 1 pour un ciel dégagé.

C.3.2 Caractérisation des sources lumineuses

Plusieurs valeurs doivent être définies. Tout d'abord un calcul d'éphémérides permet de déterminer la position du soleil (θ_s, ϕ_s) au moment et au lieu définis par l'utilisateur pour la visualisation. Ensuite nous devons calculer la valeur de la couleur ambiante qui correspond au modèle de ciel ainsi défini ainsi que la position de la source directionnelle et sa couleur.

C.3.2.1 Couleur de l'éclairage

Les modèles de ciel ne donnent que la luminance mais pas la répartition de l'énergie dans les différentes couleurs RVB. Il faut donc déterminer la couleur de l'éclairage à associer à la luminance du modèle de ciel. Afin d'améliorer le rendu au lever et au coucher du soleil, nous faisons varier la couleur de l'éclairage selon l'angle du soleil avec le sol. La couleur est interpolée linéairement entre les couleurs des angles limites. La figure C.4 présente les angles particuliers que nous avons choisi pour le changement de couleurs.

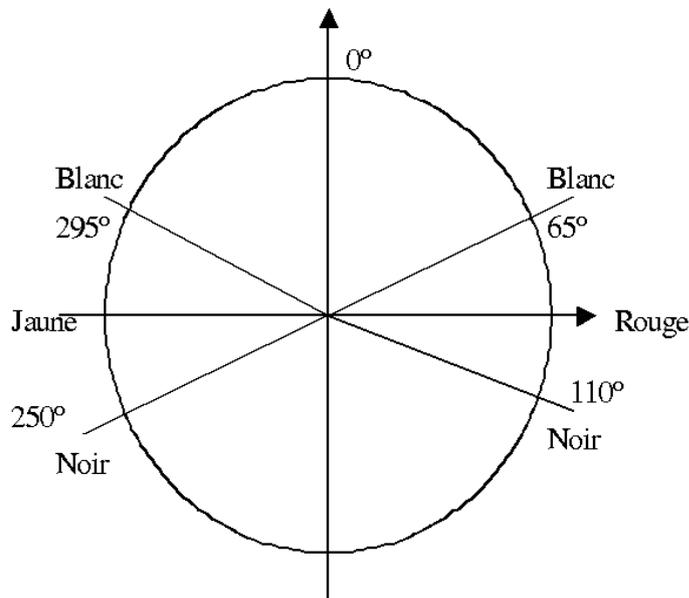


Figure C.4. *Choix de la couleur de l'éclairage en fonction de l'angle du soleil avec la terre. Dans un souci de clarté, nous utilisons des dénominations couleurs simples au lieu des triplets RVB utilisés.*

C.3.2.2 Calcul de la couleur ambiante

La couleur ambiante de la scène correspond à l'éclairage moyen du ciel. Nous choisissons donc d'intégrer le modèle de ciel afin d'obtenir une valeur moyenne de la luminance :

$$L_{moy} = \frac{1}{N} \sum_{i=1}^N L(\theta_i, \phi_i)$$

où N est le nombre d'échantillons utilisés pour le calcul de la valeur moyenne de la luminance, et $L(\theta_i, \phi_i)$ est la luminance de l'élément de ciel i .

Pour convertir la luminance L_{moy} en intensité I_{moy} , nous utilisons des valeurs de luminance minimum L_{min} et maximum L_{max} définies par l'utilisateur. Afin d'adapter au mieux l'intensité de la couleur à la perception (logarithmique) de l'œil, nous utilisons une interpolation logarithmique :

$$I_{moy} = \frac{\ln(L_{moy}) - \ln(L_{min})}{\ln(L_{max}) - \ln(L_{min})}$$

Une fois cette intensité calculée, il faut la multiplier par la couleur de l'éclairage.

C.3.2.3 Représentation du dôme du ciel

Cependant, l'effet de réalisme n'est pas seulement donné par les sources lumineuses. La portion du ciel que l'on voit modifie notre perception. Il semble alors intéressant d'inclure un hémisphère (dôme du ciel). La luminance de ce dôme est calculée à partir du même modèle de ciel que l'éclairage.

Dans un souci d'optimisation, nous choisissons de ne représenter que la partie visible du ciel par une portion de sphère fixe par rapport à la caméra. Cette portion de sphère est définie par 2500 points environ afin d'assurer une variation continue de la couleur du ciel qui est calculée aux points.

La couleur de chaque point est calculée de la manière suivante :

$$C_{point} = C_{ciel} + I_{point}C_{soleil}$$

où I_{point} est l'intensité du point, C_{ciel} est la couleur du ciel à l'horizon et C_{soleil} est la couleur du ciel dans la direction du soleil. Cette dernière pourrait être la couleur de l'éclairage définie plus haut. Mais pour obtenir une meilleure apparence du ciel, on modifie un peu cette couleur. Les deux couleurs précédentes sont calculées sur le même principe que la couleur de l'éclairage.

De la même façon que pour le ciel, nous utilisons une portion de sphère fixe par rapport à la caméra pour afficher les nuages. Les nuages sont représentés par plusieurs textures superposées. Ces textures sont plaquées sur la portion de sphère. La stratégie de plaquage est montrée sur la figure C.5.

Les coordonnées de texture d'un point du ciel sont données par l'intersection de la droite définie par la normale N en ce point et du plan P parallèle au sol et tangent à la sphère représentant le ciel, C étant la position de l'observateur. Après avoir défini les limites de la texture sur le plan P , les coordonnées x et y du point I donnent les coordonnées de texture u et v .

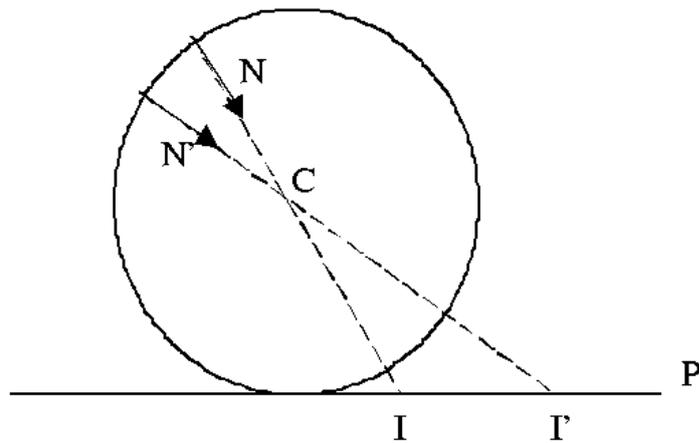


Figure C.5. *Méthode de plaquage des textures de nuages.*

Pour un point à l'horizon, la normale est parallèle au plan P. On recadre la normale entre la normale de points limites. La normale en ces points définit de plus les limites de la texture sur le plan P.

La texture appliquée au-delà des angles limites (cas où l'observateur serait en altitude) est transparente.

La figure C.6 montre les résultats obtenus.

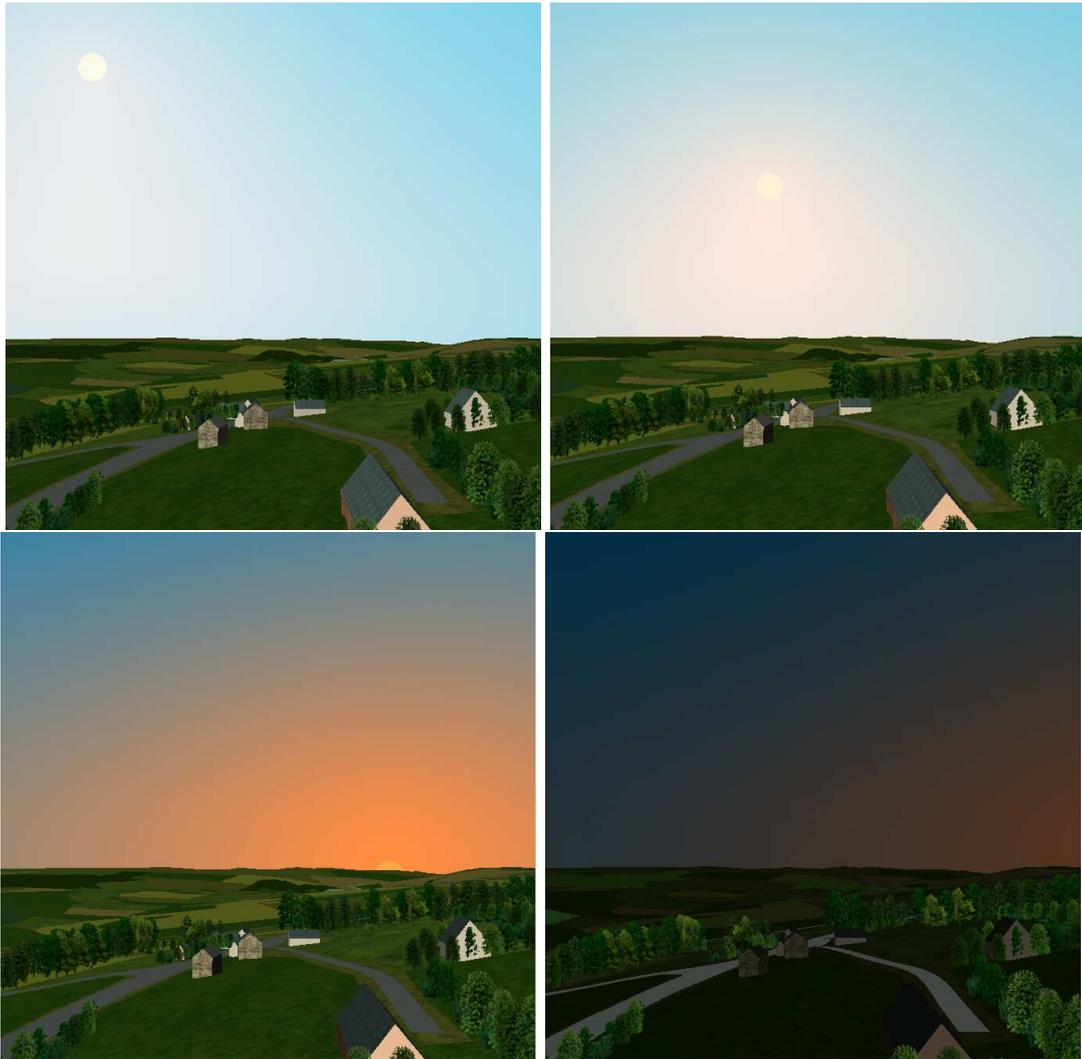


Figure C.6. *Evolution du modèle de ciel et d'éclairage.*

Bibliographie

- [AK94] James R. Arvo and David B. Kirk. Particle transport and image synthesis. In *Computer Graphics (ACM SIGGRAPH 90 Proceedings)*, volume 24, pages 63–66, August 1994.
- [AM96] D. Arquès and S. Michelin. Proximity radiosity : Exploiting coherence to accelerate form factor computations. *6th EG Workshop on Rendering*, pages 144–153, June 1996.
- [App68] A. Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the Spring Joint Computer Conference*, pages 37–45, 1968.
- [Arv86] James R. Arvo. Backward ray tracing. In *ACM SIGGRAPH'86 Course Notes - Developments in Ray Tracing*, volume 12, August 1986.
- [BB88] C. Bouville and K. Bouatouch. *Introduction to New Ray-tracing Techniques*. 1988.
- [BCW99] P. Bekaert, R. Cools, and Y. D. Willems. An empirical comparison of monte carlo radiosity algorithms. In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)*, pages 9–16, Plzen-Borey, Czech Republic, February 1999. University of West Bohemia.
- [Ben75] Jon L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9) :509–517, 1975.
- [Ben79] Jon L. Bentley. Multidimensional binary search trees in database applications. *IEEE Trans. on Soft. Eng.*, 5(4) :333–340, July 1979.
- [BG84] G. Buchsbaum and A. Gottschalk. Chromaticity coordinates of frequency-limited functions. *J. Opt. Soc. Am. A*, 1 :885–887, 1984.

- [BH83] Craig Bohren and Donald Huffman. *Absorption and scattering by a sphere*. New York : John Wiley & Sons, 1983.
- [BLS93] P. Blasi, B. LeSaëc, and C. Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum*, 12(3) :201–210, September 1993.
- [BLS94] P. Blasi, B. LeSaëc, and C. Schlick. An importance driven monte-carlo solution to the global illumination problem. *5th EG Workshop on Rendering*, pages 173–183, June 1994.
- [BPPP05] Antoine Boudet, David Pratumarty, Paul Pitot, and Mathias Paulin. Photon splatting for participating media. In *Proceedings of Graphite 2005*. ACM SIGGRAPH, December 2005.
- [BT92] N. Bhate and A. Tokuta. Photorealistic volume rendering of media with directional scattering. *3rd EG Workshop on Rendering*, pages 227–245, May 1992.
- [BW86] Geoff Wyvill Brian Wyvill, Craig McPheeters. *Soft objects*. 1986.
- [BW96] Philippe Bekaert and Yves D. Willems. *Hirad : A Hierarchical Higher Order Radiosity Implementation*. In *Proceedings of the Twelfth Spring Conference on Computer Graphics (SCCG '96)*, Bratislava, Slovakia, June 1996. Comenius University Press.
- [CCF94] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, October 1994. ISBN 0-89791-741-3.
- [CDK04] Baoquan Chen, Frank Dachille, and Arie E. Kaufman. Footprint area sampled texturing. *Visualization and Computer Graphics, IEEE Transactions on*, 10(2) :230–240, March/April 2004.
- [Cha50] S. Chandrasekar. *Radiative Transfer*. Oxford Univ. Press, 1950.
- [CIP00] Jean Claude Iehl and Bernard Péroche. An adaptive spectral rendering with a perceptual control. In *Computer Graphics Forum*, volume 19, pages 291–300, 2000.
- [Cla03] Luc Claustres. *Modélisation de la Fonction de Distribution de la Réflectance Bidirectionnelle par Ondelettes pour le Rendu Physiquement Réaliste*. PhD thesis, Université Paul Sabatier (Toulouse), 2003.

- [Col79] Munsell Color. *Munsell Book of Color*. Macbeth a division of Kollmorgen Corp., 1979.
- [Col94] Steven Collins. Adaptive splatting for specular to diffuse light transport. In *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, July 1994.
- [CPC84] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In *Computer Graphics (ACM SIGGRAPH'84 Proceedings)*, pages 137–145, July 1984.
- [CSSD96] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Global Illumination of Glossy Environments Using Wavelets and Importance. *ACM Transactions on Graphics*, 15(1) :37–71, January 1996.
- [CT82] Robert L. Cook and Kenneth E. Torrance. A reflection model for computer graphics. *ACM TOG*, 1(1) :7–24, January 1982.
- [CW91] Hong Chen and En-Hua Wu. Radiosity for furry surfaces. In *Eurographics'91*, pages 447–457, September 1991.
- [CW93] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, MA, 1993.
- [DF92] Mark Drew and Brian Funt. Natural metamers. *CVGIP : Image Understanding*, 56(2) :139–151, September 1992.
- [DMCP94] Pascal M. Deville, Slimane Merzouk, Dorothee Cazier, and Jean C. Paul. Spectral data modeling for a lighting application. *Computer Graphics Forum*, 13(3) :97–106, September 1994.
- [Dom74] G.A. Domoto. Frequency integration for radiative transfer problems involving homogeneous non-gray gases : the inverse transmission function. *Journal of Quantitative Spectroscopy and Radiative Transfer.*, 14 :935–942, 1974.
- [Dut96] Philip Dutré. *Mathematical Frameworks and Monte Carlo Algorithms for Global Illumination in Computer Graphics*. PhD thesis, University of Leuven, September 1996.
- [Dut01] Philip Dutré. Global illumination compendium. *Cornell University*, 2001.
- [D'Z92] M. D'Zmura. Color constancy : Surface color from changing illumination. *Opt. Soc. Am. A*, 9 :7–24, 1992.

- [FDF94] Graham D. Finlayson, Mark S. Drew, and Brian V. Funt. Color constancy : Generalized diagonal transformation. *J. Opt. Soc. Am. A*, 11 :3011–3019, 1994.
- [Fom04] Boris A Fomin. A k-distribution technique for radiative transfer simulation in inhomogeneous atmosphere : 1. fkdm, fast k-distribution model for the longwave. *Journal of Geophysical Research : Atmospheres (1984–2012)*, 109(D2), 2004.
- [For57] G. E. Forsythe. Generation and use of orthogonal polynomials for data fitting with a digital computer. *J. Soc. Ind. Appl. Math.*, 5 :57, 1957.
- [For90] D. A. Forsyth. A novel algorithm for color constancy. *Int. J. Comput. Vision*, 5 :5–36, 1990.
- [GBNHAR98] Antonio Garcia-Beltran, Juan L. Nieves, Javier Hernandez-Andres, and Javier Romero. Linear bases for spectral reflectance functions of acrylic paints. *Color Res. Appl.*, 23 :39–45, 1998.
- [GCS93] S. Gortler, M. F. Cohen, and P. Slusallek. Radiosity and relaxation methods. Technical Report Technical Report 1/1993, Computer Science Department, University of Erlangen-Nuremberg, 1993.
- [GH96] Simon Gibson and R. J. Hubbard. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5) :297–310, December 1996.
- [GHJ96] Robert Geist, Oliver Heim, and Stephen Junkins. Color representation in virtual environments. *Color Res. Appl.*, 21(2) :121–128, April 1996.
- [GK96] Allen Van Gelder and Kwansik Kim. Direct volume rendering with shading via three-dimensional textures. In *1996 ACM/IEEE Symposium on Volume Visualization*, San Francisco, October 1996.
- [Gla95] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.
- [GSCH93] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 221–230, 1993.
- [GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the Interaction of Light Between Diffuse Surfaces. In *Computer*

- Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 212–222, July 1984.
- [Hal89] Roy A. Hall. Illumination and color in computer generated imagery. *Springer-Verlag*, 1989.
- [HDS99] J. M. Hasenfratz, C. Domez, F. Sillion, and G. Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. In *Computer Graphics Forum (Proc. Eurographics '99)*, volume 18, pages C–221–C–232, September 1999.
- [Hec89] Paul Heckbert. Fundamentals of texture mapping and image warping. Master's thesis, University of California, 1989.
- [Hec90] Paul Heckbert. Adaptive radiosity textures for bidirectional ray tracing. volume 24, pages 145–154, August 1990.
- [Hec93] Paul Heckbert. Introduction to Finite Element Methods. In *ACM SIGGRAPH '93 Course Notes - Global Illumination*, chapter 3, pages 1–10. 1993.
- [HG83] Roy A. Hall and Donald P. Greenberg. A testbed for realistic image synthesis. *IEEE Computer Graphics & Appl*, 3(6) :10–20, November 1983.
- [Hor84] Berthold K. P. Horn. Exact reproduction of colored images. *Computer Vision, Graphics, and Image Processing*, 26 :135–167, 1984.
- [HSA91] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4) :197–206, July 1991.
- [ICG86] Dave S. Immel, Michael Cohen, and Donald P. Greenberg. A Radiosity Method for Non-Diffuse Environments. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, pages 133–142, August 1986.
- [IK66] E. Isaacson and H. B. Keller. Analysis of numerical methods. *John Wiley and Sons*, 1966.
- [Jen95] Henrik Wann Jensen. Global illumination using photon maps. *Eurographics Rendering Workshop*, pages 326–335, June 1995.
- [Jen01] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A K PETERS LTD edition, 2001.
- [Kaj86] James T. Kajiya. The rendering equation. *Computer Graphics, Proc. of ACM SIGGRAPH 86*, 20(4) :143–150, 1986.

- [Kay88] S. M. Kay. Modern spectral estimation : Theory and application. *Prentice-Hall, Englewood Cliffs, New Jersey*, 1988.
- [KH84] J.T. Kajiya and B.P. Von Herzen. Ray tracing volume densities. *Computer Graphics*, 18(3) :165–174, July 1984.
- [Kri53] E. L. Krinov. Spectral reflectance properties of natural formations. *National Research Council of Canada, Technical Translation TT-439, translated by G. Belkov*, 1953.
- [Lav03] Fabien Lavignotte. *Calcul de l'éclairage global par estimation de densité et par une approche image*. PhD thesis, Université Paul Sabatier - Toulouse III, July 2003.
- [LBC94] E. Languéno, K. Bouatouch, and M. Chelle. Global illumination in presence of participating media with general properties. *5th EG Workshop on Rendering*, pages 69–85, June 1994.
- [LC03] Bent Dalgaard Larsen and Niels Jorgen Christensen. Optimizing photon mapping using multiple photon maps for irradiance estimates. *WSCG POSTER proceedings*, February 2003.
- [LDMP89] M. Lucas, P.Martin D. Martin, and D. Plemenos. In *Journées GROPLAN*, 1989.
- [LH91] David Laur and Pat Hanrahan. Hierarchical splatting : A progressive refinement algorithm for volume rendering. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 285–288, July 1991.
- [LP03] Fabien Lavignotte and Mathias Paulin. Scalable Photon Splatting for Global Illumination. In *Graphite 2003 (International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia)*, Melbourne, Australia., pages 1–11. ACM SIGGRAPH, 11-14 février 2003.
- [LSG94] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and Error Estimates for Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 67–74, 1994.
- [LTG92] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Discontinuity Meshing for Accurate Radiosity. *IEEE Computer Graphics and Applications*, 12(6) :25–39, November 1992.

- [LTG93] D. Lischinski, F. Tampieri, and D. P. Greenberg. Combinig hierarchical radiosity and discontinuity meshing. In *Proc. of SIGGRAPH-93 : Computer Graphics*, pages 199–208, Anaheim, CA, 1993.
- [LW93] Eric P. Lafortune and Yves D. Willems. Bidirectional path tracing. *Compugraphics '93*, pages 95–104, 1993.
- [LW96] E.P. Lafortune and Y. Willems. Rendering participating media with bidirectional path tracing. *6th EG Workshop on Rendering*, pages 92–101, June 1996.
- [Mal86] Laurence T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3(10) :1673–1683, October 1986.
- [Max94] N.L. Max. Efficient light propagation for multiple anisotropic volume scattering. *5th EG Workshop on Rendering*, pages 57–104, June 1994.
- [Mey88] Gary W. Meyer. Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41 :57–79, 1988.
- [Mie08] Gustav Mie. Beiträge zur optik trüber medien, speziell kolloidaler metalllösungen. *Annalen der Physik*, 25 :377–445, 1908.
- [MMI⁺98] K. Mueller, T. Moller, J.E. Swan II, R. Crawfis, N. Shareef, and R. Yagel. Splatting errors and antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, 4(2) :178–191, April–June 1998.
- [MMS⁺04] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis and rendering of bidirectional texture functions. In *Eurographics 2004 State of the Art Reports*, pages 69–94, 2004.
- [Moo45] P. Moon. Polynomial representation of reflectance curves. *J. Opt. Soc. Am.*, 35 :597–600, 1945.
- [MW86] Laurence T. Maloney and Brian A. Wandell. Color constancy : A method for recovering surface spectral reflectance. *J. Opt. Soc. Am. A*, 3 :29–33, 1986.
- [MW92] David Marimont and Brian Wandell. Linear models of surface and illuminant spectra. *J. Opt. Soc. Am. A*, 9 :1905–1913, 1992.
- [MY96] Klaus Mueller and Roni Yagel. Fast perspective volume rendering with splatting by utilizing a ray-driven approach. In *IEEE Visualization '96*. IEEE, October 1996. ISBN 0-89791-864-9.

- [Mys97] Karol Myszkowski. Lighting reconstruction using fast and adaptive density estimation techniques. pages 251–262, 1997.
- [NDN96] Tomoyuki Nishita, Y. Dobashi, and Eihachiro Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. *Computer Graphics, Proc. of ACM SIGGRAPH 96*, pages 379–386, 1996.
- [NNJ43] Sidney M. Newhall, Dorothy Nickerson, and Deane B. Judd. Final report of the o. s. a. subcommittee on the spacing of the munsell chip. *J. Opt. Soc. Am.*, 33 :385–418, 1943.
- [NRH⁺77] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. *Geometric Considerations and Nomenclature for Reflectance*. Monograph 161, National Bureau of Standards(US), October 1977.
- [Pat93] C. Patmore. Simulated multiple scattering for cloud rendering. *Graphics, Design and Visualization*, pages 59–70, 1993.
- [Pee93] Mark S. Peercy. Linear color representations in full spectral rendering. *Computer Graphics, Proc. of SIGGRAPH 93*, pages 191–198, 1993.
- [PKK00] M. Pauly, T. Kollig, and A. Keller. Metropolis light transport for participating media. *Proceedings of the Eurographics Workshop on Rendering*, pages 11–22, June 2000.
- [PM92] S.N. Pattanaik and S.P. Mudur. Computation of global illumination by monte carlo simulation of the particle model of light. In *Third Eurographics Workshop on Rendering*, pages 71–83, 1992.
- [PM93] S.N. Pattanaik and S.P. Mudur. Computation of global illumination in a participating medium by monte carlo simulation. *The Journal of Vis. and Comp. Animation*, 4(3) :133–152, 1993.
- [PW93] Donald B. Percival and Andrew T. Walden. Spectral analysis for physical applications. *Cambridge Univesity*, 1993.
- [RF91] Maria Raso and Alain Fournier. A piecewise polynomial approach to shading using spectral distributions. *Graphics Interface 91*, pages 40–46, 1991.
- [RGBHA97] Javier Romero, Antonio Garcia-Beltran, , and Javier Hernandez-Andre. Linear bases for representation of natural and artificial illuminants. *J. Opt. Soc. Am.*, 14(5) :1007–1014, May 1997.

- [RP97] Gilles Rougeron and Bernard Peroche. A adaptive representation of spectral data for reflectance computations. *Proc. of the 8th Eurographics Workshop on Rendering*, pages 127–138, 1997.
- [RT87] Holly Rushmeier and K. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics, Proc. of ACM SIGGRAPH 87*, 21(4) :293–302, 1987.
- [SAWG91] Francois Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A Global Illumination Solution for General Reflectance Distributions. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 187–196, July 1991.
- [SB97] W. Stuerzlinger and R. Bastos. Interactive Rendering of Globally Illuminated Glossy Scenes. In *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 93–102, June 1997.
- [SDF99] Yinlong Sun, Mark S. Drew, and F. David Fracchia. Representing spectral functions by a composite model of smooth and spiky components for efficient full-spectrum photorealism. *IEEE Workshop on Photometric Modeling of Computer Vision and Graphics*, pages 4–11, June 1999.
- [SFD98] Yinlong Sun, F. David Fracchia, and Mark S. Drew. A composite model for representing spectral functions. *Simon Fraser University, Technical Report SFU CMPT TR 1998-18*, 1998.
- [SG69] J. Spanier and M. Gelbard. Addison-Wesley, Reading, MA, 1969.
- [Sil86] B. W. Silverman. Density estimation for statistics and data analysis. *Chapman and Hall, London and New York*, 1986.
- [Sil94] Francois Sillion. Clustering and Volume Scattering for Hierarchical Radiosity Calculations. In *Fifth Eurographics Workshop on Rendering*, pages 105–117, Darmstadt, Germany, June 1994.
- [Sil95] François X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3), September 1995.
- [Smi94] Brian Smits. *Efficient Hierarchical Radiosity in Complex Environments*. Ph.D. thesis, Technical Report, Ithaca, NY, 1994.

- [SSSS98] M. Stamminger, H. Schirmacher, P. Slusallek, and H.-P. Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Journal (Proc. Eurographics '98)*, 17(3) :C165–C174, September 1998.
- [Sta95a] J. Stam. *Multi-Scale Stochastic Modelling of Complex Natural Phenomena*. PhD thesis, Dept. of Computer Science, University of Toronto, 1995.
- [Sta95b] J. Stam. Multiple scattering as a diffusion process. *6th EG Workshop on Rendering*, pages 41–50, 1995.
- [Sta00] Marc Stamminger. *Finite Element Methods for Global Illumination Computations*. PhD thesis, Universitat Erlangen, Nurnberg, Germany, 2000.
- [SW00] Frank Suykens and Yves D. Willems. Density control for photon maps. *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, 2000.
- [SWH⁺95] Peter Shirley, Bretton Wade, Philip M. Hubbard, David Zareski, Bruce Walter, and Donald P. Greenberg. Global illumination via density estimation. pages 219–230, 1995.
- [SWO77] W. S. Stiles, G. Wyszecki, and N. Ohta. Counting metameric object-color stimuli using frequency-limited spectral reflectance function. *J. Opt. Soc. Am.*, 67 :779–784, 1977.
- [Vea97] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, December 1997.
- [VG94] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. *Fifth Eurographics Workshop on Rendering*, pages 147–162, 1994.
- [VG97] Eric Veach and Leonidas J. Guibas. Metropolis light transport. *SIGGRAPH 97 Proceedings*, pages 65–76, August 1997.
- [VGI94] Michael J. Vrhel, Ron Gershon, and Lawrence S. Iwan. Measurement and analysis of object reflectance spectra. *Color Res. Appl.*, 19 :4–9, 1994.
- [VMKE00] V. Volevich, K. Myszkowski, A. Khodulev, and Kopylov E.A. Using the Visible Differences Predictor to Improve Performance of Progressive Global Illumination Computations. *ACM Transactions on Graphics*, 19(2) :122–161, 2000.
- [VT92] Michael J. Vrhel and H. J. Trussel. Color correction using principle components. *Color Res. Appl.*, 17 :328–338, 1992.

- [Wal98] Bruce Johnathan Walter. *Density Estimation Techniques for Global Illumination*. PhD thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, August 1998.
- [Wes89] Lee Westover. Interactive volume rendering. In *Proceedings of the Chapel Hill Workshop on Volume Visualization*, pages 9–16, May 1989.
- [Wes90] Lee Westover. Footprint evaluation for volume rendering. *Computer Graphics*, 24(4) :367–376, August 1990.
- [Whi80] Turner Whitted. An improved illumination model for shaded display. *Communications of the Association for Computing Machinery*, 23 :343–349, 1980.
- [WRC88] Greg Ward, Francis Rubinstein, and Robert Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of SIGGRAPH 1988*, volume 22, pages 85–92, 1988.
- [Yui87] A. Yuille. A method for computing spectral reflectance. *Biol. Cybern.*, 56 :195–201, 1987.
- [Zeg97] Eric Zeghers. *Radiosité spectrale et sources directionnelles : application à l'éclairage des tunnels routiers*. PhD thesis, Université de Rennes 1, Rennes, FRANCE, 1997.
- [Zie89] O. C. Zienkiewicz. *The Finite Element Method, 4th edition*. McGrawHill, 1989.
- [ZPVG01] Matthias Zwicker, Hanspeter Pfister, J. VanBaar, and M. Gross. EWA volume splatting. In *IEEE Visualization 2001*, 2001.

Résumé

LANCER DE PHOTONS MULTI-PASSES ET ÉCRASEMENT DE PHOTONS POUR LE RENDU OPTRONIQUE

Mots-clés: Synthèse d'image, optronique, infrarouge, diffusion, illumination globale, lancer de photons, écrasement de photons, milieu participant, estimation de densité

La simulation de l'éclairage par illumination globale a fait l'objet de nombreuses recherches et applications au cours des dernières années. Tout d'abord utilisée dans le domaine visible, la simulation est aujourd'hui de plus en plus appliquée au rendu infrarouge. On appelle *optronique* l'union de ces deux domaines. Le problème principal des méthodes d'illumination globale actuelles provient de la difficulté à traiter le phénomène de diffusion de la lumière, aussi bien dans le cas des surfaces que des milieux participants. Ces méthodes offrent des résultats satisfaisants dans le cas de scènes simples, mais les performances s'effondrent lorsque la complexité augmente. Dans la première partie de cette thèse, nous exposons la nécessité de la prise en compte des phénomènes de diffusion pour la simulation optronique. Dans la deuxième partie nous posons les équations qui unifient les différentes méthodes de synthèse d'image, c'est-à-dire l'équation du rendu et l'équation volumique du transfert radiatif. L'état de l'art des méthodes d'illumination globale présenté dans la troisième partie montre qu'à l'heure actuelle la méthode des cartes de photons est celle qui offre le meilleur compromis performance/qualité. Néanmoins, la qualité des résultats obtenus grâce à cette méthode est dépendante du nombre de photons qui peuvent être stockés et donc de la quantité de mémoire disponible. Dans la quatrième partie de la thèse, nous proposons une évolution de la méthode, *le lancer de photons multi-passes*, qui permet de lever cette dépendance mémoire, et ainsi d'obtenir une très grande qualité sans pour autant utiliser une configuration matérielle onéreuse. Un autre problème de la méthode des cartes de photons est le temps de calcul important nécessaire lors du rendu de milieux participants. Dans la cinquième et dernière partie de cette thèse, nous proposons une méthode, *l'écrasement de photons volumique*, qui prend avantage de l'estimation de densité pour reconstruire efficacement la luminance volumique à partir de la carte de photons. Notre idée est d'isoler le calcul de la

diffusion et d'utiliser une approche duale de l'estimation de densité pour l'optimiser car il constitue la partie coûteuse du calcul. Bien que les temps de rendu obtenus par notre méthode sont meilleurs que ceux obtenus en utilisant la méthode des cartes de photons pour la même qualité, nous proposons aussi une optimisation de la méthode utilisant les nouvelles capacités des cartes graphiques.

Abstract

MULTIPASS PHOTON MAPPING AND PHOTON SPLATTING FOR OPTRONIC RENDERING

Keywords: Image synthesis, optronic, infrared, scattering, global illumination, photon mapping, photon splatting, participating media, density estimation

Much research have been done on global illumination simulation. Firstly used in the visible spectrum domain, today, simulation is more and more applied to infrared rendering. The union of these two domains is called *optronic*. The main problem of the current global illumination methods comes from the complexity of the light scattering phenomena, as well for surfaces as for participating media. These methods offer satisfactory results for simple scenes, but performances crash when complexity raises. In the first part of this thesis, we expose the necessity to take scattering phenomena into account for optronic simulation. In the second part, we pose the equations that unify all global illumination methods, i.e. the rendering equation and the volume radiative transfer equation. The state of the art presented in the third part shows that the Photon Mapping method is, at this moment, the one that offers the better compromise between performance and quality. Nevertheless, the quality of the results obtained with this method depends on the number of photons that can be stored, and then on the available memory. In the fourth part, we propose an evolution of the method, called *Multipass Photon Mapping*, which permits to get rid of this memory dependency, and hence, to achieve a great accuracy without using a costly hardware configuration. Another problem inherent to Photon Mapping, is the enormous rendering time needed for participating media rendering. In the fifth and last part of this thesis, we propose a method, called *Volume Photon Splatting*, which takes advantage of density estimation to efficiently reconstruct volume radiance from the photon map. Our idea is to separate the computation of emission, absorption and out-scattering from the computation of in-scattering. Then we use a dual approach of density estimation to optimize this last part as it is the most computational expensive. Our method extends Photon Splatting, which optimizes the computation time of Photon Mapping for surface rendering, to participating media, and then considerably reduce participating media rendering times. Even though our method is faster than Photon Mapping for equal quality, we also propose a GPU based optimization of our algorithm.

