Contrôle Automatique de Qualité pour l' Éclairage Global

Xavier Granier

▶ **To cite this version:**

Xavier Granier. Contrôle Automatique de Qualité pour l' Éclairage Global. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2001. Français. NNT : . tel-00524825v1

HAL Id: tel-00524825

https://theses.hal.science/tel-00524825v1

Submitted on 8 Oct 2010 (v1), last revised 30 Nov 2010 (v2)

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

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

Université Grenoble I - Joseph Fourier

Contrôle Automatique de Qualité pour l'Éclairage Global

Xavier GRANIER

Thèse présentée pour l'obtention du titre de :
Docteur de l'Université Joseph Fourier
Spécialité Informatique
Arrêté ministériel du 5 juillet 1984 et du 30 mars 1992
Préparée au sein du laboratoire :
*i*MAGIS-GRAVIR/IMAG-INRIA (UMR CNRS C5527)
Soutenue le 9 Novembre 2001

Composition du jury :

Joelle	COUTAZ	Présidente
Bernard	PEROCHE	Rapporteur
Peter	SHIRLEY	Rapporteur
Christophe	SCHLICK	
Georges	DRETTAKIS	Directeur de thèse
Claude	PUECH	Co-directeur de thèse

À Jean Granier
Grand amatteur de sciences et surtout de lettres

À Mathieu Perrot
À qui la thèse a volée son parain.

Remerciements

Voila une page qui me tient particulièrement à cœur, et je m'en voudrais de l'avoir oublié. En effet, ce document n'aurait jamais pu avoir lieu sans le soutien de nombreuses personnes. Je tiens donc à les remercier à travers ces lignes.

Je tiens en premier lieu à remercier Georges, pour son soutien de chaque instant durant ces trois années (et plus si il est tenu compte du DEA). Il a su me donner confiance en mes travaux, ce qui fut loin d'être toujours facile ! Il m'a permis de découvrir, par les longues discussions que nous avons eues, des orientations nouvelles. Il a été un conseiller et un support, c'est ce que j'attendais d'un directeur de thèse.

Je tiens aussi à remercier profondément Claude pour m'avoir accepté au sein de l'équipe *i*MAGIS, et pour tout le travail qu'il y effectue. Cette équipe, par sa diversité de personnes et la volonté de recherche qui y règne est un endroit privilégié pour conduire un projet de recherche.

Je tiens de plus à remercier les membres du jury, pour avoir accepter cette charge et cette responsabilité. Leur questions et leur remarques m'ont offert un nouvel aperçu sur mes travaux.

Il est pour moi indispensable d'inclure dans mes remerciements l'ensemble des membres d'*i*MAGIS, passés et présents, pour leur accueil et leur amitié. Ils constituent une véritable équipe. Quitter cette équipe est une étape difficile du fait des amitiés nouées en son sein. Je tiens particulièrement à remercier Cyrille pour son soutien dans les problèmes administratifs.

Dans ces remerciements, la famille tiens une place à part. Elle est bien sûr le plus vieux soutien que je connaisse, aussi bien mes parents que mes frères. Elle reste pour moi le lieu d'échange principal, et je ne pourrais jamais les remercier assez. Il ne faudrait surtout pas que j'oublie Odile et Roger, pour avoir accepter de m'herberger cette dernière année.

Les personnes que j’ai rencontrées au cours de ma vie ont toutes eu toute une influence sur ce que je suis, mais certaines plus que d’autres. Je tiens à remercier particulièrement pour leur amitié Benjamin et Lionel qui sont devenus pour moi comme des frères, mais aussi Anne-Claire, Benoît, Marie-Sophie et Simon (les deux), pour cette extraordinaire aventure humaine à travers le projet scout en Égypte. Je veux aussi citer ici Eddie, Olivier, Gabriel, Thomas, Eric, Nicolas, ainsi que David et Clo, qui m’ont accompagné de leur amitié.

Ces remerciements ne sont pas exhaustifs et je pris ceux que j’ai oublié de citer au moment d’écrire ces lignes de me pardonner.

Sommaire

1 Introduction **1**

2 État de l'art **7**
1 Principes généraux 7
2 Les méthodes déterministes 12
3 Les méthodes stochastiques 21
4 Qualité d'un éclairage 28

3 Plateforme d'expérimentation **33**
1 Architecture logicielle : G.I.S. 34
2 Traitement hiérarchique 36
3 Raffinement 40
4 Lissage 43
5 Conclusion 44

4 Nouvelle méthode unifiée **47**
1 Introduction 47
2 Aperçu général de la méthode 48
3 Intégration des deux algorithmes 52
4 Accélération de la simulation 57
5 Reconstruction et rendu 58
6 Résultats 59
7 Conclusion 61

5 Mise à jour incrémentale **65**
1 Introduction 65
2 Reconstruction des caustiques par textures 68
3 Lancer de particules incrémental 75
4 Cas diffus 78
5 Affichage interactif de la solution 79
6 Résultats 81
7 Conclusion 85

6 Reconstruction finale **89**
1 Présentation générale 90
2 Reconstruction de la composante diffuse 90
3 Reconstruction et intégration des caustiques 95
4 Composante dépendante du point de vue 97
5 Premiers résultats 101
6 Conclusion 105

7 Conclusion **107**

Annexes **109**

A Paramétrisation des objets **113**
1 Sphère et portion de sphère 113
2 Cône et portion de cône 116
3 Cylindre et portion de cylindre 119
4 Anneau et portion d'anneau 122

B Volume d'interaction interactif **125**

C Description des librairies **127**

1

Introduction

Contexte

Depuis plus de vingt ans, de nombreux efforts ont été consacrés à l'augmentation du réalisme des images générées par un ordinateur. Dans ce but, une grande partie de la recherche a été orientée vers la résolution des phénomènes d'éclairage global. Le calcul de la juste répartition de la lumière dans l'environnement accroît de manière significative le réalisme de l'image, grâce d'une part aux effets lumineux créés, et mais aussi par l'ambiance lumineuse qui en résulte.

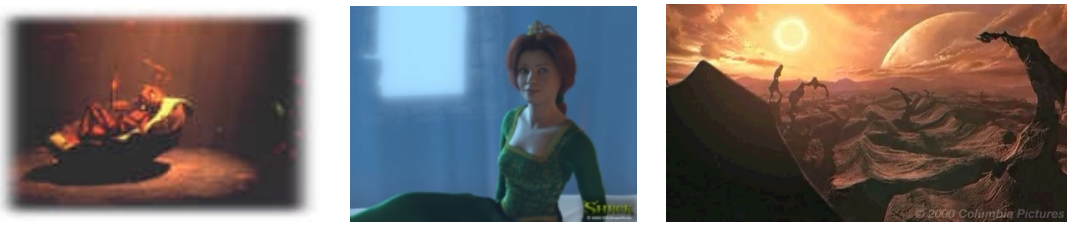
À l'heure actuelle, la plupart des problèmes ont étés identifiés (phénomènes de diffusion, reflets lumineux, caustiques, modifications interactives, gestion de la complexité, et bien d'autres), et la plupart ont déjà reçu une solution dans certains cadres particuliers. Il est ainsi possible de générer tous les effets lumineux souhaités.

De nombreuses solutions proposées reposent sur la résolution de l'équation de propagation de la lumière. Les méthodes numériques qui en résultent (déterministes ou probabilistes) permettent ainsi la simulation des effets d'éclairage global. Il est possible aujourd'hui de faire une simulation sur la plupart des modèles, en prenant le temps et les ressources nécessaires.

Pour certains phénomènes, comme les caustiques, des approches plus spécifiques, concentrées sur la manière d'obtenir l'effet souhaité plutôt que sa simulation physique, permettent de générer localement des phénomènes d'apparence globale.

Ainsi, grâce à l'ensemble des méthodes qui ont été développées au cours de ces vingt dernières années, il est possible d'obtenir des images d'un très grand réalisme (cf. Fig. 1.1). Mais celles-ci restent souvent des exploits, et elles sont en général présentées comme cela.

En effet, derrières ces images se cache le travail de nombreuses personnes, travaillant des journées entières à ajuster les algorithmes et à combiner les différents



Fourmi'Z (©DreamWorks SKG) | Shrek (©DreamWorks SKG) | Final Fantasy (©SquareUsa)

FIG. 1.1 : Images tirées de trois films récents

résultats partiels, afin d'obtenir au mieux les effets d'éclairage global et le réalisme souhaité.

Motivations

Dans ce contexte, il est plus que nécessaire de fournir les outils permettant d'automatiser le travail requis pour l'obtention d'images réalistes, au sens de l'éclairage. Il faut réduire les étapes nécessaires à l'heure actuelle afin de permettre un meilleur contrôle sur la qualité finale. A travers la figure Fig. 1.2, différentes compostions sont montrées. Si certains effets sont ignorées, une partie du réalisme ce perd. Mais combiner différents algorithmes chacun indépendant peut devenir laborieux, comme le choix des méthodes pouvant aboutir à un tel résultat.

En effet, la recherche de solutions efficaces pour la simulation de chaque phénomène lumineux a conduit à une explosion du nombre de méthodes, chacune étant plus ou moins spécialisée. Il est donc difficile de choisir et de composer les méthodes pour le calcul d'une image comprenant tous les effets lumineux souhaités. Le premier problème réside donc en l'*automatisation* de la composition de tous les phénomènes dans un outil permettant ainsi de contrôler d'une manière simplifiée l'ensemble des effets globaux. La composition réduite diminue d'autant les difficultés de contrôle sur la qualité finale d'une image.

De plus, ce contrôle doit aussi passer par la possibilité de prévisualiser rapidement un aperçu de ce que sera la solution finale. Avec cet aperçu, il devient plus simple d'ajuster les paramètres et d'affiner les modèles afin d'obtenir les effets souhaités. Le calcul doit donc être rapide, voire même interactif, même si le prix à payer est une dégradation de qualité de l'image.

D'une manière générale, nous voulons fournir un outil simplifiant le contrôle sur le résultat final, et bien sûr, sur sa qualité, ceci dans le contexte de l'éclairage global. Nous allons donc commencer par étudier l'intégration de méthodes existantes.

Contributions

A travers ce document, nous allons démontrer qu'une telle intégration est possible, et qu'elle permet d'offrir un meilleur contrôle sur la qualité finale d'une image. Elle se base principalement sur le partitionnement de l'espace des transferts lumineux. Par le positionnement de tels échanges (appelés *liens* dans un algorithme de radiosité hiérarchique), il est possible de détecter des chemins lumineux plus complexes, et ainsi obtenir rapidement des effets complexes comme des caustiques. Nous présenterons comment tirer le plus d'information possible de cette structure, afin d'accélérer et d'automatiser les calculs.

Cette démonstration va passer par le développement d'un algorithme unifié, combinant un algorithme de radiosité hiérarchique avec regroupement et une méthode de lancer de particules. Le lancer de particules sera déterminé par l'utilisation des liens, restreignant l'espace d'émission possible et accélérant ainsi le processus. Cette utilisation permet aussi une détection automatique des configurations générant des effets directionnels. Cette intégration permet une bonne variation entre une solution rapide et approximative et une solution plus lente mais de grande qualité. Elle se base sur le partitionnement des échanges lumineux grâce à l'usage des liens présents dans l'algorithme hiérarchique. Cette structure permet de répartir automatiquement les méthodes intégrées afin de les employer dans leurs domaines de prédilection.

Nous montrerons qu'une telle méthode permet une certaine interaction avec la solution d'éclairage globale, permettant ainsi un contrôle sur le modèle dans le but d'obtenir les effets souhaités. Pour cela nous développerons une spécialisation de cet algorithme, par une limitation des calculs nécessaires, basée ici aussi sur l'utilisation des liens, restreignant ici l'espace de calcul. Cette approche permet une mise à jour rapide et locale pour des scènes complexes.

De plus, nous nous attacherons à développer des techniques permettant d'améliorer la reconstruction finale, que cela soit pour une visualisation interactive, avec l'emploi de reconstruction alternative de la fonction d'éclairage, ou pour le calcul d'une image de grande qualité, en se basant sur les informations déjà obtenues par la solution. Encore une fois, le partitionnement effectué dans les étapes précédentes va permettre de concentrer les efforts dans les zones où cela se révèle nécessaire.

Organisation du document

Avant d'aborder nos contributions, nous présenterons un tour d'horizon assez général des méthodes existantes, en nous attardant plus longuement sur les méthodes déterministes (radiosité hiérarchique avec regroupement) et probabilistes, et notamment sur le lancer de particules. En effet, notre algorithme repose sur ces deux approches. Ce rapide tour d'horizon est aussi l'occasion de poser les bases d'une réflexion sur le contrôle de qualité.

Puis, par la suite, nous aborderons la mise en place de notre plateforme d'expérimentation, base du développement de notre algorithme (cf. Chap. 3). Cette mise en

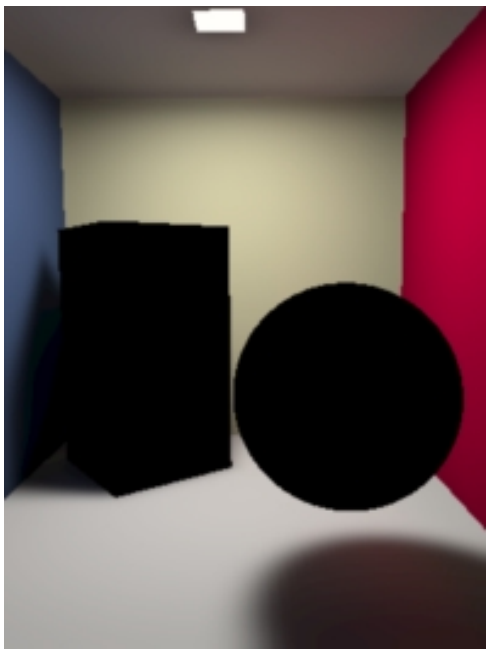
place permet de proposer des solutions à des problèmes spécifiques, comme le regroupement, le traitement des objets autres que polygonaux, le raffinement, le calcul de la visibilité et du facteur de forme.

Ensuite, nous présenterons notre algorithme unifié, combinant un seul outil, une méthode de radiosité hiérarchique avec regroupement et un lancer de particules (cf. Chap. 4). Cette méthode permet de calculer une solution complète vers les surfaces diffuses, et ce, en autorisant une variation entre un calcul rapide, de qualité moyenne, et un calcul de bien plus grande qualité. Nous proposerons aussi une méthode pour l'affichage interactif de reflets sur une telle solution.

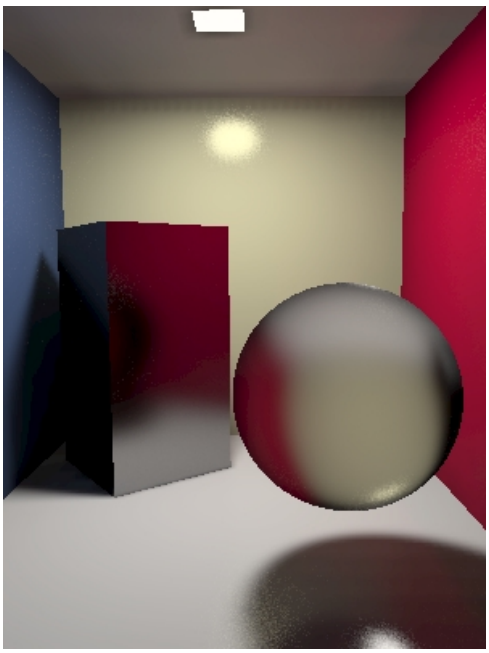
A partir de cet algorithme, nous présenterons dans le chapitre suivant une technique permettant d'obtenir des mises à jour rapides, voire interactives, pour des solutions d'éclairage global (cf. Chap. 5). Cela permet d'agir sur le positionnement des objets, pour obtenir les effets désirés, sans avoir à tout recalculer. Nous présenterons aussi une technique de reconstruction séparant les détails diffus de ceux dus à un éclairage indirect hautement directionnel.

Nous présenterons en dernière contribution, une méthode permettant, à partir de la solution calculée par notre algorithme unifié, d'obtenir des images de très grande qualité (cf. Chap. 6).

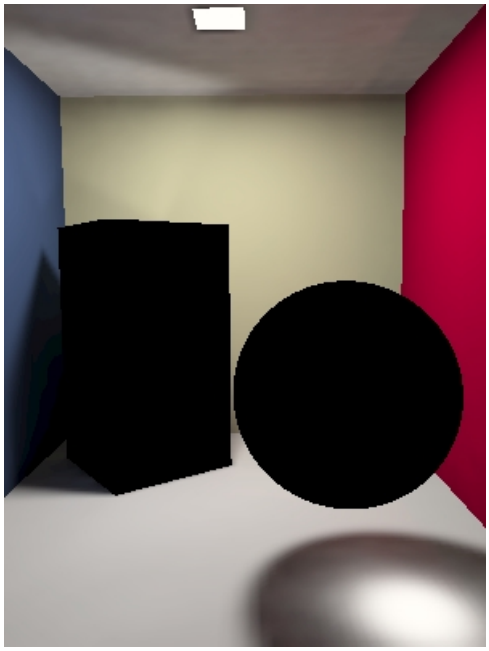
Enfin, nous conclurons par un résumé des contributions et par des pistes de recherches futures permettant d'étendre ces travaux, avec toujours comme objectif principal une possibilité accrue de contrôle automatique sur la qualité.



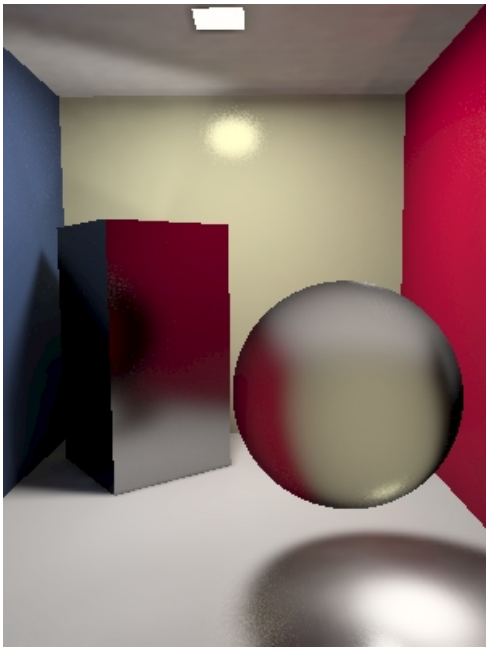
diffus



diffus + reflets



diffus + effets directionnels



diffus + effets directionnels + reflets

FIG. 1.2 : Composition des effets lumineux

2

État de l'art

Depuis près de vingt ans maintenant, de nombreux efforts se sont portés sur la création d'algorithmes efficaces dans la simulation des échanges lumineux pour l'image de synthèse. Les algorithmes présentés dans cette thèse étant une intégration de plusieurs approches, un tour d'horizon de ces techniques est plus que nécessaire. Nous présenterons ici les principes généraux de l'éclairage global, les approches déterministes, majoritairement représentées par les méthodes de radiosité, ainsi que les approches statistiques.

1 Principes généraux

Cette section a pour but d'introduire le formalisme général pour toute méthode d'éclairage global. Nous y présenterons les notions de base nécessaires au développement de notre sujet[1]. Elles constituent le cadre général de toute approche de l'éclairage global.

1.1 Ce que l'on perçoit

Vouloir créer des images réalistes, c'est vouloir tromper notre vision, c'est donner l'illusion à notre œil que ce qu'il voit est réel. Or la perception visuelle du monde réel est liée à *l'énergie lumineuse* que reçoivent les cônes est les bâtonnets, structures réceptrices de notre rétine.

Il semble donc maintenant naturel que les premières formulations pour la simulation de l'éclairage en synthèse d'images, apparues dans les années 1980 *[GTGB84, Kaj86]*, se soient basées sur des travaux de thermodynamique, plus précisément sur les

[1] De nombreuses références et équations sont disponibles dans *[Dut96]* et *[Dut]*

études d'échanges thermiques*[Wie66, SC78, SH81, How82]*. L'idée générale qui dirige ces algorithmes d'éclairage global est l'équilibre en régime stationnaire des échanges lumineux (considérés sous la seule forme énergétique) en un point de l'espace et pour une longueur d'onde donnée.

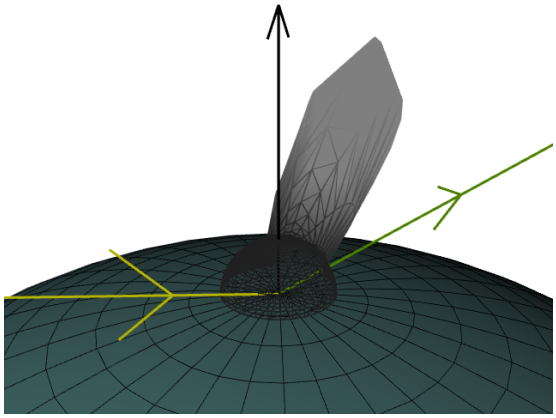
La résolution de cet équilibre sur la rétine ou en chaque point d'une scène donnée permet l'obtention des effets lumineux globaux, augmentant ainsi le réalisme d'un environnement virtuel.

1.2 Équation du rendu : équilibre lumineux

Formulée pour la première fois par Kajiya en 1986 *[Kaj86]*, l'équation du rendu est une écriture de cet équilibre. Elle ne marque pas le début de l'éclairage global, mais elle est une généralisation sous la forme d'une équation bilan de ce qui avait été fait jusqu'alors. Elle sous-entend simplement que l'énergie émise par une portion de surface est la somme de l'énergie propre de la surface et de la réémission d'une partie de l'énergie reçue, ceci pour une longueur d'onde donnée. Dans cette hypothèse, les effets de transfert d'énergie d'une longueur d'onde vers une autre (comme la fluorescence) sont négligés. L'équation peut alors être écrite sous la forme :

$$L(s,\omega_o) = L_P(s,\omega_o) + \int_\Omega \rho(s,\omega_i,\omega_o) L_i(s,\omega_i) < \omega_i \cdot n > d\omega_i \qquad (2.1)$$

s	=	point sur la surface S
ω_o	=	direction sortante
ω_i	=	direction incidente
Ω	=	ensemble des directions
n	=	normal en s
ρ	=	fonction de transmission
L	=	énergie émise
L_P	=	énergie propre émise
L_i	=	énergie incidente
$< \cdot >$	=	produit scalaire



La quantité ainsi considérée, appelée **radiance**, représente la quantité d'énergie émise par une portion de surface et dans une direction donnée ($W.m^{-2}.sr^{-1}$, voir table 2.1)

Dans le cas où le point s' de la scène 3D est parfaitement visible du point s, alors l'énergie reçue en s est l'énergie émise en s', soit, $L_i(s,\omega) = L(s',\omega)$. De plus, la relation entre la variation d'angle solide $d\omega$ et la portion de surface ds' se présente sous la forme (cf. Fig. 2.1) :

$$d\omega = \frac{< -\omega \cdot n' >}{||s'-s||^2} ds' \text{ avec } \omega = \frac{s'-s}{||s'-s||} \qquad (2.2)$$

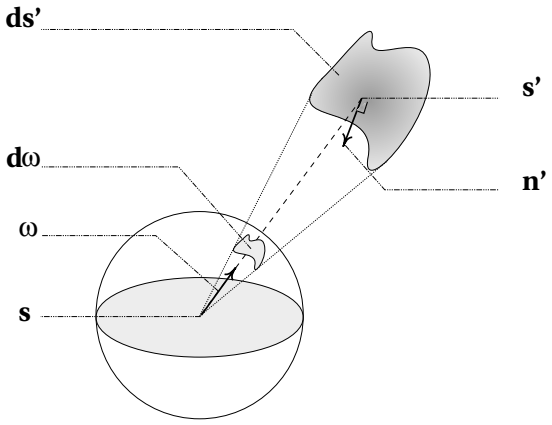


FIG. 2.1 : Angle solide

L'équation 2.1 ne dépend alors plus que de la radiance émise par les surfaces et devient :

$$L(s,\omega_o) = L_P(s,\omega_o) + \int_{\mathbb{S}} \rho(s,\omega,\omega_o) L(s',\omega) V(s,s') \frac{<-\omega \cdot n'><\omega \cdot n>}{\|s'-s\|^2} ds' \quad (2.3)$$

De nouveaux termes ont été introduits, $\mathbb{S}$ pour l'ensemble des points de la scène, et $V(s,s')$, la visibilité entre deux de ces points. Cette fonction vaut 1 lorsque les deux points (s et s') sont parfaitement visibles entre eux, 0 s'il existe un obstacle entre les deux. Dans le cas d'un milieu absorbant, cette fonction peut prendre toute les valeurs dans $[0,1]$.

L'équation 2.1 suppose connue la propriété de réflexion de la surface ρ, couramment appelée **BRDF** *[NRH+77]* (pour *"Bidirectionnal Reflection Distribution Function"* en anglais) ou pour les réfractions **BTDF** (*"Bidirectionnal Transmission Distribution Function"*), ou la combinaison de tous les types de réémission possibles, **BSDF** (*"Bidirectionnal Scattering Distribution Function" [Hec91]*). Cette fonction représente la répartition directionnelle de l'énergie réémise. Sa grandeur est donc en sr^{-1} (voir table 2.1). La conservation de l'énergie impose que $\int_\Omega \rho(s,\omega,\omega_o) < \omega \cdot n > d\omega \leq 1$, ceci quelle que soit la direction ω_o considérée. Elle est donc assimilable à une fonction de densité en probabilité.

1.3 Définition d'un flux

Parmi les grandeurs utiles, le flux permet de bien représenter la propagation de l'énergie lumineuse dans un milieu, et surtout c'est une grandeur facilement mesurable. Elle représente une puissance lumineuse seule (W - voir table 2.1) émise ou reçue d'une surface.

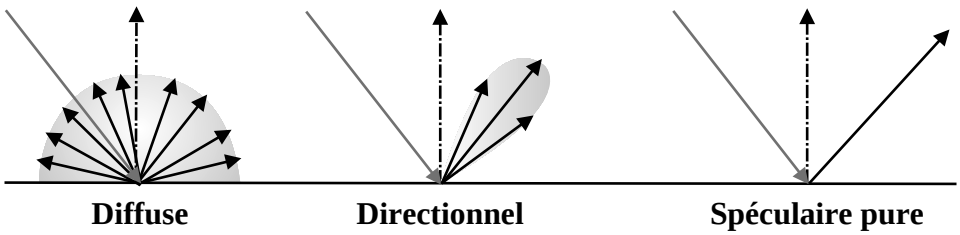


FIG. 2.2 : Différent types de propriété de réflexion

Le flux peut être ainsi considéré dans deux directions : l'émission ou la réception d'énergie. Le flux quittant un élément S peut être décrit par l'intégrale :

$$\Phi_S^{s+} = \int_S \int_{\Omega^+(s)} L(s,\omega) < \omega \cdot n > d\omega ds \tag{2.4}$$

où $\Omega^+(s)$ représente le demi-espace des directions possibles en s dans la direction de la normale n et $\Omega^-(s)$ l'autre demi-espace. De même, le flux reçu par une surface S s'exprime sous la forme :

$$\Phi_S^{i+} = \int_S \int_{\Omega^+(s)} L_i(s,\omega) < \omega \cdot n > d\omega ds \tag{2.5}$$

soit, de même que pour l'équation 2.3,

$$\Phi_S^{i+} = \int_S \int_{\mathbb{S}} L(s',\omega) V(s,s') \frac{< \omega \cdot n >< -\omega \cdot n' >}{\|s-s'\|^2} d\omega ds' \tag{2.6}$$

1.4 Séparation des types de réflexion

Pour caractériser les différents types de réflexion présents dans la nature, deux catégories sont en général distinguées : **diffuse** et **directionnelle**.

Pour un matériau diffus, la direction de réflexion n'a pas d'influence. Ainsi, l'énergie incidente est redistribuée de manière égale dans toutes les directions. La BSDF (voir section 1.2) ne dépend donc plus des directions incidentes et sortantes ($\rho(s,\omega,\omega_o) = \rho(s)$).

Tous les autres types de retransmission sont dits *directionnels*. L'exemple extrême de ce type de matériau est celui du miroir, où il n'y a réflexion que dans une unique direction de retransmission, celle donnée par la loi de Descartes.

De par la nature additive de la lumière, il est alors possible de définir une BSDF quelconque comme étant la somme d'une composante diffuse $\rho_d(s)$ et d'une composante directionnelle $\rho_s(s,\omega,\omega')$:

$$\rho(s,\omega,\omega') = \rho_d(s) + \rho_s(s,\omega,\omega') \tag{2.7}$$

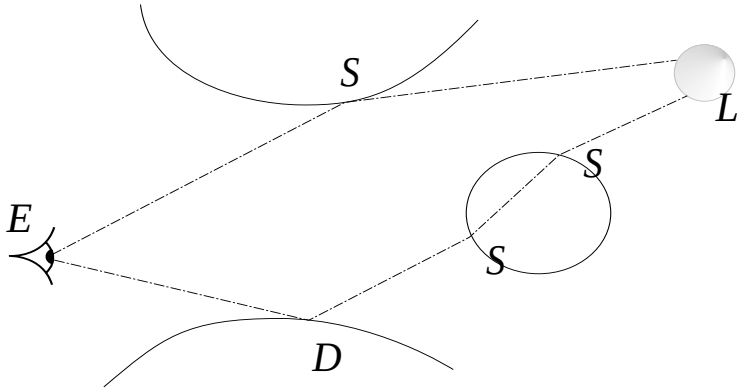


FIG. 2.3 : Des chemins lumineux

Ainsi, il est possible de caractériser le chemin suivi par un photon à travers un milieu par les différents types de réflexion qu'il a subi. Heckbert a introduit *[Hec90]* une caractérisation de ces chemins par une expression régulière. Les sources de lumière sont représentées par L, une réflexion diffuse par D, une spéculaire par S et enfin, le récepteur (l'œil par exemple) par E. Ainsi, tous **les chemins lumineux** sont caractérisés par l'expression régulière $L(D|S)^*E$ (cf. Fig. 2.3). Le signe "*" représente zéro ou plusieurs retransmissions et "|", l'opérateur "ou".

1.5 Hypothèse diffuse : Nouvelle formulation

Dans le cadre d'une scène purement diffuse, la BRDF est alors constante par rapport aux directions, soit $\rho(s,\omega,\omega_o) = \rho(s)$. L'équation du rendu Eq. 2.3 peut ainsi s'écrire :

$$L(s,\omega_o) = L_P(s,\omega_o) + \rho(s)\int_{\mathbb{S}} L(s',\omega)V(s,s')\frac{<-\omega\cdot n'><\omega\cdot n>}{\|s'-s\|^2}ds' \qquad (2.8)$$

Une nouvelle grandeur, indépendante de la direction d'émission, la **radiosité** *[How82, GTGB84]* est alors introduite, notée en général *B*. Elle correspond au flux de lumière émis par unité de surface. Sa grandeur est donc en Watt par mètre carré ($W.m^{-2}$).

$$B(s) = \int_{\Omega^+(s)} L(s,\omega) < \omega \cdot n > d\omega \qquad (2.9)$$

Dans un environnement purement diffus, où les sources de lumière et les matériaux sont diffus, l'énergie émise par une surface est invariante suivant la direction ($L(s,\omega) = L(s)$). Dans ce cas, la radiosité est proportionnelle à la radiance d'un facteur π :

$$B(s) = \pi L(s) \qquad (2.10)$$

Symbole	Dénomination	Quantité
L	radiance	$W.m^{-2}.sr^{-1}$
B	radiosité	$W.m^{-2}$
I	irradiance	$W.m^{-2}$
Φ	flux	W
ρ	BSDF	sr^{-1}
R	Coefficient de réflexion	$-$
A	Aire	m^2

TAB. 2.1 : Les différentes grandeurs

De même, un nouveau coefficient de réflexion R_d est défini, représentant la proportion totale d'énergie incidente réémise diffusément :

$$R_d(s) = \int_{\Omega^+(s)} \rho(s) < \omega \cdot n > d\omega = \pi\rho(s) \tag{2.11}$$

Ce coefficient est donc sans grandeur (cf. Tab. 2.1).

L'équation du rendu 2.3 peut ainsi être réécrite dans le cas diffus, en fonction de la seule radiosité :

$$\begin{cases} B(s) &= B_P(s) + R_d(s)I(s) \\ I(s) &= \int_{\mathbb{S}} B(s')V(s,s')\frac{<-\omega \cdot n'><\omega \cdot n>}{\pi\|s'-s\|^2}ds' \end{cases} \tag{2.12}$$

La grandeur $I(s)$, appelée **irradiance**, représente le flux total d'énergie reçue par une portion de surface dS'. Sa variation est bien plus faible que celle de l'énergie émise comme cela a été montré par Ward *[WH92]*, car intégrée sur l'ensemble des directions d'incidence.

Dans cette équation, un autre terme apparaît, terme purement géométrique (cf. Eq. 2.13), représentant la proportion d'énergie émise par une portion de surface qui est reçue par une autre. Ce terme est souvent nommé noyau de l'équation de radiosité.

$$\begin{cases} G(s,s') &= V(s,s')g(s,s') \\ g(s,s') &= \frac{<-\omega \cdot n'><\omega \cdot n>}{\pi\|s'-s\|^2} \end{cases} \tag{2.13}$$

2 Les méthodes déterministes

Les premières solutions déterministes d'éclairage global, ont fait l'hypothèse d'un environnement purement diffus *[GTGB84]*. Cela permet la simplification du problème : le résultat est indépendant du point de vue, la fonction à calculer ne dépend alors plus que des points de la scène et non plus de la direction, réduisant ainsi la dimension du problème. La discrétisation peut alors se faire simplement sur l'espace *[SP94]*.

2.1 Discrétisation : hypothèse diffuse

Supposons donnée une discrétisation de la scène en e éléments $(E_i, i = 1..e)$ (avec e, nombre d'éléments), sur chacun desquels la fonction de radiosité va être évaluée dans une base orthogonale de fonctions $(b_i^j(s), j = 1..k)$ (avec k, nombre de fonctions de base) :

$$\forall i \in [1..e], B_i(s) = \sum_{j=1}^{k} B_i^j b_i^j(s) \tag{2.14}$$

La fonction de radiosité sur un élément E_i est alors représentée par un vecteur de coefficients $(B_i^j, j = 1..k)$. Les coefficients B_i^j sont les projections de la fonction de radiosité i sur la base $(b_i^j(s), j = 1..k)$ de l'élément :

$$B_i^j \int_{E_i} (b_i^j(s))^2 ds = \int_{E_i} B_i(s) b_i^j(s) ds \tag{2.15}$$

En projetant l'équation 2.12 sur les bases de fonctions des éléments de discrétisation :

$$B_i^j = (B_p)_i^j + \frac{1}{\int_{E_i} (b_i^j(s))^2 ds} \sum_{l=1}^{e} \sum_{m=1}^{k} B_l^m \int_{E_i} R_d(s) \int_{E_l} G(s,s') b_i^j(s) b_l^m(s') ds ds' \tag{2.16}$$

En se plaçant dans un cadre constant, c'est à dire une seule fonction de base constante par élément ($b_i(s) = 1$ sur E_i et 0 ailleurs), l'intégrale $\int_{E_i} (b_i^1(s))^2 ds$ devient $\int_{E_i} ds$, ce qui représente l'aire de l'élément de discrétisation E_i, soit A_i. De plus, en supposant également que le coefficient de réflexion est constant par élément, la formulation 2.15 se simplifie et devient

$$B_i = (B_p)_i + (R_d)_i \sum_{l=1}^{e} B_l \frac{1}{A_i} \int_{E_i} \int_{E_l} G(s,s') ds ds' \tag{2.17}$$

Il apparaît alors ici un terme purement géométrique, décrivant la proportion de l'énergie émise par un élément E_j qui est reçue par l'élément E_i. Il est nommé **facteur de forme** *[Spa63, How82]* $F_{i,j}$:

$$F_{i,l} = \frac{1}{A_i} \int_{E_i} \int_{E_l} G(s,s') ds ds' \tag{2.18}$$

Comme les facteurs de forme expriment une proportion d'énergie émise, leur sommation à partir d'un élément doit être inférieure ou égale à 1 :

$$\sum_{i=1}^{N} F_{i,j} \leq 1 \tag{2.19}$$

L'équation de radiosité 2.12 s'exprime à présent matriciellement :

$$B_p = MB$$

$$M = \begin{bmatrix} 1-(R_d)_1F_{1,1} & -(R_d)_1F_{1,2} & . & . & -(R_d)_1F_{1,n} \\ -(R_d)_2F_{2,1} & 1-(R_d)_2F_{2,2} & . & . & . \\ . & . & . & . & . \\ . & . & . & 1-(R_d)_{n-1}F_{1,n-1} & (R_d)_{n-1}F_{1,n-1} \\ (R_d)_nF_{n,1} & . & . & (R_d)_nF_{n,n-1} & 1-(R_d)_nF_{n,n} \end{bmatrix} \tag{2.20}$$

Du fait de la relation 2.19, M est à diagonale dominante (strictement ou non, cela dépend de la propriété matérielle de réflexion moyenne de la scène). La méthode de Gauss-Seidel est donc généralement employée pour la résolution, mais les algorithmes de relaxation (Southwell) et de gradient conjugué *[GTGB84]* sont aussi envisageables.

2.2 Facteur de forme et visibilité

Le cœur du problème consiste ici à calculer les facteurs de forme. Les traiter dans leur globalité, avec la visibilité, se révèle complexe dans la grande majorité des configurations. Cela n'est possible en général qu'en utilisant un échantillonnage par lancer de rayon *[WEH89, HSA91, TNSP98]* pour le calcul de l'intégrale, ou par une discrétisation des directions possibles et une reprojection des objets de la scène dans cet espace ("hémicube" *[CG85]*).

Comme il existe des configurations géométriques pour lesquelles un calcul exact du facteur de forme est possible, sans prendre en compte la visibilité, *[How82, Spa63, SC78, GTGB84, BRW89, MBG86]*, et que ce calcul peut être étendu dans le cadre non diffus pour des miroirs plans *[WCG87]* et des surfaces spéculaires *[SP89]*, il est courant de séparer l'évaluation en deux termes (cf. Eq. 2.18 et Eq. 2.13), afin de simplifier le calcul et de pouvoir apporter une réponse adaptée à chaque cas :

$$\begin{cases} F_{i,j} &= V_{i,j}\frac{1}{A_i}\int_{E_i}\int_{E_j} g(s,s')\,ds\,ds' \\ V_{i,j} &= \frac{1}{A_iA_j}\int_{E_i}\int_{E_j} V(s,s')\,ds\,ds' \end{cases} \tag{2.21}$$

Pour évaluer le second terme $V_{i,j}$, de nombreux travaux ont été menés pour tenter de résoudre le problème de la visibilité entre deux éléments d'une scène 3D. Des études théoriques *[TH94, DDP97, Dur99]* ont permis la mise en place d'un cadre nécessaire au calcul d'une visibilité exacte, d'autres de trouver les discontinuités de visibilité *[Hec91, SLD92, OF99]*, offrant la possibilité de créer un maillage basé sur ces discontinuités *[LTG92, LTG93a, DF94, Stu94, DS96, GS96, DDP99]*.

Mais toutes ces méthodes restent coûteuses en temps et en mémoire, et elles manquent de robustesse. Des solutions approximatives sont communément choisies. Elles sont en général basées soit sur un échantillonnage et un lancer de rayons *[HSA91,*

$\Delta B = B_p$ // *radiosité à transmettre de chaque élément (shooting)*
$B = 0$ // *radiosité totale de chaque élément*
Tant que $\|B\| \geq$ *erreur* **faire**
soit j l'indice du $Max(\Delta B)$: alors $\Delta Rad = \Delta B_j$ et $\Delta B_j = 0$.
pour tous les éléments *e* de la scène **faire**

$$\begin{cases} B_i = B_i + (R_d)_i F_{i,j} \Delta Rad \\ \Delta B_i = \Delta B_i + (R_d)_i F_{i,j} \Delta Rad \end{cases}$$

FIG. 2.4 : Algorithme de radiosité progressive (cas de l'émission progressive)

Ng96], soit sur l'utilisation de cartes d'ombres *[Sol98]*, soit sur le remplacement des obstacles par un milieu absorbant *[SD95]*.

2.3 Radiosité progressive

Pour éviter le coût de stockage en $O(N^2)$ et une complexité de calcul en $O(N^3)$, l'algorithme de radiosité progressive *[CCWG88, BRW89, Che90]* ne considère qu'un seul émetteur, le plus puissant, à chaque itération. Il se base ainsi sur le principe de réflexion successive. A chaque étape, un seul élément transmet sa radiosité et stocke celle qu'il va recevoir en retour (figure 2.4). Il a été démontré *[WH97b]* que la convergence de ces algorithmes se fait en $O(N^2)$.

Avec une telle formulation, la matrice des facteurs de forme n'a plus besoin d'être stockée, car une seule ligne est utilisée par itération. Le coût de stockage devient alors en $O(N)$, mais il est nécessaire de recalculer les N facteurs de forme à chaque itération. De plus, cet algorithme permet une approche plus simple pour un raffinement adaptatif, suivant le gradient de radiosité, par exemple, pour obtenir une meilleure représentation de détails comme les ombres.

Le principal désavantage de cette méthode est la difficulté de contrôler l'erreur finale, du fait du choix d'un émetteur parmi plusieurs possibles à chaque itération. Certains éléments peuvent n'avoir ainsi aucune influence, car ils auront toujours une radiosité à émettre trop faible. Un terme de lumière d'ambiance est en général rajouté pour palier cette imprécision. De même, lorsqu'une source est composée d'un nombre important de polygones, la convergence sera lente du fait de l'émission élément par élément, problème en partie corrigé avec les méthodes par groupe *[Kok93, RR99]*.

2.4 Radiosité hiérarchique avec regroupement

A travers les méthodes précédentes, l'intérêt du choix d'une stratégie de subdivision du maillage s'est affirmée pour une bonne représentation de la fonction de radiosité. Certains lieux de la scène demandent une représentation plus fine pour une meilleure approximation de la fonction de radiosité. La radiosité progressive a montré l'avantage d'un tel maillage adaptatif (*voir ci-dessus*), mais cette méthode reste trop

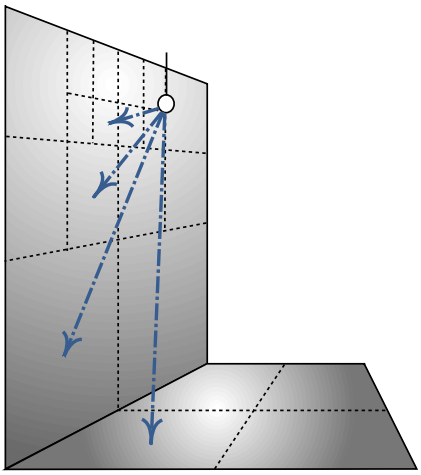


FIG. 2.5 : Quatre échanges à des niveaux différents

rigide, car seuls les éléments non-subdivisés peuvent être émetteurs et il n'y a pas de maîtrise de l'éclairage secondaire. Une représentation multirésolution des échanges permet, elle, d'apporter en plus du maillage adaptatif, un contrôle de l'erreur globale par un contrôle de l'erreur locale.

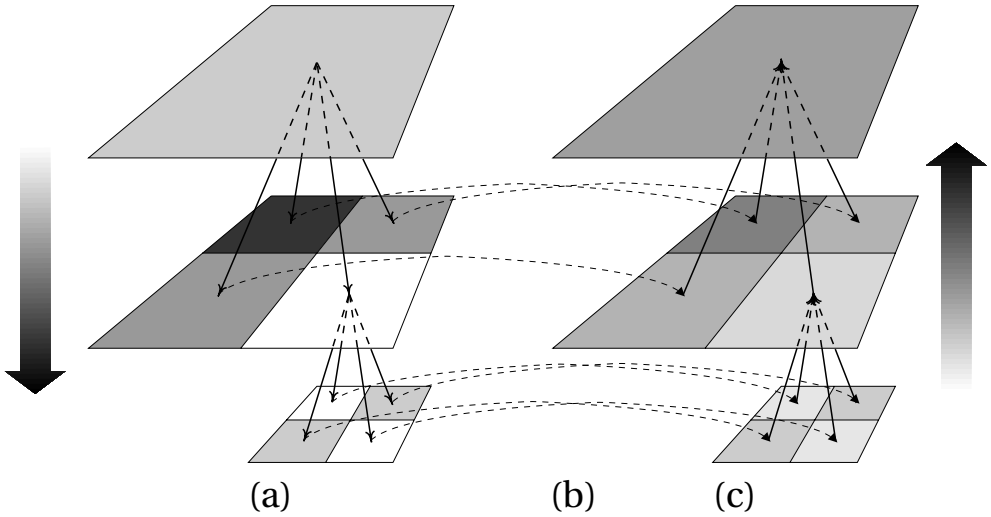
Tirant parti de cette représentation, la radiosité hiérarchique *[HSA91]* est l'une des méthodes les plus rapides pour résoudre l'équation de radiosité.

Principes

Les différences d'importance qui existent entre l'influence des divers objets de la scène dans l'éclairage d'un objet sont ici exploitées, ceci dans le but d'accélérer le calcul. La radiosité hiérarchique tient compte de ces différences et ne simule un échange qu'au "juste" niveau de hiérarchie. C'est une représentation multirésolution des échanges et de la radiosité. Les feuilles de la hiérarchie sont la base de discrétisation de la fonction de radiosité.

Des liens entres les éléments permettent d'établir les échanges énergétiques. Ils conservent des informations comme le facteur de forme pour ne pas avoir à le recalculer. Ces liens modélisent l'interaction entre un émetteur et un récepteur. Ils sont stockés habituellement sur les récepteurs et stockent les émetteurs (*figure 2.5*). Le calcul de radiosité pour chaque itération se fait en trois étapes *[HSA91, SP94]* qui peuvent êtres distinctes ou intégrées *[GD99]* :

- **Raffinement** : Création de la hiérarchie et positionnement des liens au "juste niveau" d'échange. Le critère de raffinement est alors ce "juste niveau". Il peut être basé sur l'irradiance apportée par l'émetteur ($F_{i,j}B_j$), l'énergie apportée ($A_iF_{i,j}B_j$), sur des calculs de variations *[GH96]*, ou bien, sur des bornes d'erreurs *[LSG94]*.
- **Collecte d'énergie** (*Gather*) : sur chaque élément de la hiérarchie, l'irradiance reçue à ce niveau est calculée.



(a) Descente et accumulation de l'irradiance (b) Réflexion diffuse (c) Mise à jour de la représentation multirésolution (moyenne dans le cas constant)

FIG. 2.6 : Détail de la mise à jour hiérarchique

– **Mise à jour hiérarchique** (*Push-Pull*, cf. Fig. 2.6) : un parcours récursif de la hiérarchie permet de sommer l'ensemble de l'irradiance reçue par les feuilles de la hiérarchie. Sur ces feuilles, l'irradiance y est réfléchie afin d'y calculer la valeur de radiosité. Enfin, la représentation multirésolution de la fonction de radiosité est mise à jour.

Afin de faciliter la présentation détaillée de ces trois étapes, les notations suivantes sont définies :

– $\mathbb{H}$: Ensemble des éléments de la hiérarchie
– $\mathbb{L}_i$: Ensemble des éléments liés à l'élément i (i est considéré comme un récepteur)

Algorithme

L'algorithme hiérarchique nécessite l'utilisation de deux grandeurs pour chaque élément de la hiérarchie : l'irradiance I (énergie reçue) et la radiosité B (énergie émise). Toute la radiosité est recalculée à chaque itération ($B^{i+1} = B_p + R_d F B^i$), pour chaque élément, ce qui évite une accumulation des erreurs comme en radiosité progressive.

A l'initialisation de la hiérarchie, seules les sources de lumière possèdent une radiosité qui n'est pas nulle. Après le raffinement, la mise à jour des valeurs de radiosité se déroule comme décrit ci-dessous (décomposition de la Collecte et du Push-Pull *[HSA91]*) :

– Collecte (*Gather*) : l'irradiance reçue sur chaque élément de la hiérarchie (i) est calculée, grâce aux liens :

$$I_i = \sum_{j \in \mathbb{L}_i} F_{i,j} B_j \tag{2.22}$$

– Descente (*Push*) : L'irradiance reçue par chaque feuille de la hiérarchie est calculée, soit, en descendant la hiérarchie :

$$\forall j \in \mathbb{H}, I_{fils_j} = I_{fils_j} + I_j \tag{2.23}$$

– Sur les feuilles l de la hiérarchie : Mise à jour de la radiosité de feuilles :

$$B_l = R_d I_l \tag{2.24}$$

– Calcul de la valeur "moyenne" (*pull*) : Mise à jour de la radiosité de chaque élément i, ceci en remontant la hiérarchie :

$$B_i = \frac{\sum_{j \in fils_i} A_j B_j}{\sum_{j \in fils_i} A_j} \tag{2.25}$$

La complexité de cet algorithme est en $O(k^2 + n)$, k désignant le nombre initial d'éléments, et n le nombre d'éléments après subdivision.

Le regroupement ("clustering")

Cependant la complexité reste quadratique en nombre de polygones initiaux, limitant ainsi la complexité des scènes qu'il est possible de traiter. Le regroupement *[SAG94, Sil95, CLSS97, WH97a]* propose une extension à la radiosité hiérarchique pour tenir compte des échanges à un niveau supérieur, c'est à dire entre objets hiérarchiques, pouvant être une surface (subdivisée ou non) ou un *ensemble* d'autre objets hiérarchiques. Alors que jusqu'à maintenant nous ne considérions que des surfaces, nous allons supposer maintenant qu'à un certain niveau, un volume (ou "cluster") est suffisant pour approcher le comportement des échanges. La hiérarchie devient ainsi totale, la scène étant incluse dans un cluster racine. La complexité de l'algorithme devient en $O(C^2 + N)$, N étant le nombre d'éléments initiaux de la scène, et C le nombre d'éléments fils de la racine.

En considérant que le comportement face aux échanges radiatifs d'un groupe de surfaces est isotrope, il peut être remplacé par une boîte englobante simulant son comportement global. Avec le tableau de correspondance 2.2, la formule de calcul des facteurs de forme, présentée dans les sections précédentes, reste similaire *[Sil95]* :

$$F_{i,j} = F_j F_i \frac{1}{A_i} \int_{E_i} \int_{E_j} \frac{\tau R_i(s) E_j(s')}{\pi \|s - s'\|^2} ds ds' \tag{2.26}$$

Les symboles sont définis dans la table Tab. 2.2.

nom	symbole	surface	volume
Facteur en réception	$R(s)$	$<\omega\cdot n>$ (cf. Eq. 2.12)	1
Facteur en émission	$E(s')$	$<-\omega\cdot n'>$ (cf. Eq. 2.12)	1
Facteur d'aire	A_i	A	${}_4\kappa V$
Échelle	F	1	$\frac{1}{4}$

κ = coefficient d'extinction

TAB. 2.2 : Extension du facteur de forme *[Sil95]*

Le regroupement apporte de nombreux avantages dont la réduction de la complexité des échanges, ou la simulation de milieux absorbants. D'autres approches ont été proposées, basées sur des niveaux de détail sur les surfaces polygonales, permettant ainsi de conserver une information concernant l'orientation générale *[WHG99, Dum96]*. Il a aussi été étendu aux environnement qui ne sont pas seulement isotropes *[SDS95, CLSS97]*.

Le raffinement

Le raffinement a pour but d'ajuster la discrétisation de la fonction de radiosité, mais aussi d'établir les liens (échanges énergétiques) au "juste niveau". Ce dernier est souvent défini par une minimisation d'une erreur locale et globale *[LSG94, GH96]*. Lorsque le critère de raffinement devient inférieur à un certain seuil - contrôlant ainsi l'erreur locale - un lien est établi.

Les critères de raffinement sont très variés, se basant sur une valeur unique (par exemple sur le facteur de forme F, sur l'irradiance BF, sur l'énergie reçue BFA) *[HSA91, TS01]* ou sur un intervalle de confiance (Valeur maximale - Valeur minimale) *[GH96, SSS97]*, mais toujours avec l'objectif de réduire l'erreur d'approximation par la minimisation de ces valeurs.

2.5 Reconstruction finale "Final Gather"

Souvent, le défaut d'une telle solution réside dans la taille du maillage nécessaire, maillage qui ne s'adapte pas forcément bien aux détails fins perceptibles dans une image, même lorsque celui-ci essaie de s'y conformer au maximum (*[LTG93b]*).

Afin d'accroître la qualité des images ainsi calculées, il est souvent nécessaire d'effectuer une reconstruction finale, pour chaque pixel. Cette reconstruction permet, en se basant sur la solution déjà calculée, de faire un lancer de rayons efficace *[BDW98, CLSS97]*.

La méthode la plus récente *[SSS01]*, se base sur un calcul de l'erreur faite en chacun des points du maillage et en chacun des éléments de discrétisation, afin de déterminer au plus juste les zones où un échantillonnage est nécessaire pour accroître la qualité d'une image. Cela permet de corriger les erreurs de discrétisation. Une solution fine ne nécessitera qu'une courte étape de reconstruction finale.

2.6 État des approches déterministes

Les algorithmes de radiosité, et en particulier la radiosité hiérarchique avec regroupement, se sont imposés comme des solutions efficaces pour le problème de l'éclairage global en environnement diffus. Leurs principaux attraits sont un partitionnement des échanges lumineux (par les liens en radiosité hiérarchique), un calcul de la solution indépendant du point de vue, permettant ainsi un affichage efficace à l'aide du matériel graphique pour une visualisation interactive ("walkthrough").

Ces algorithmes ont de plus été assez vite adaptés au cas dynamique *[Che90]*. En effet, il est plus aisé de localiser les calculs nécessaires pour mettre à jour une solution dans un cadre déterministe. Cela est d'autant plus vrai avec la radiosité hiérarchique avec regroupement, où l'espace des échanges se trouve partitionné par les liens. Une organisation hiérarchique de ces liens *[DS97, Sha97, GD99]* permet de détecter rapidement les zones où un nouveau calcul est nécessaire.

Depuis l'apparition des méthodes de radiosité, plusieurs approches ont été proposées pour prendre en compte les transferts non diffus. La première est de stocker la représentation directionnelle de l'énergie sortant de chaque élément de discrétisation de la solution. Des premières approches *[ICG86]* aux plus récentes *[SDS95, SSG+99, SSG+00]* utilisant les groupes et les ondelettes, le coût de stockage nécessaire rend ces méthodes généralement inutilisables pour des scènes complexes contenant des effets fortement directionnels. Les représentations continues *[SAWG91]* nécessitent elles aussi beaucoup de coefficients de base. Pour éviter ce surcoût mémoire, les méthodes en trois points *[AH93]* ont été proposées. Cependant, celles-ci conservent une complexité de stockage en k^3 liens où k est le nombre d'éléments de la scène. Ainsi, elles sont en général limitées à des scènes relativement simples. Dumont et al. *[DBG99]* ont réduit la mémoire nécessaire en un coût quadratique, par un algorithme progressif basé sur l'utilisation de la hiérarchie de liens et la suppression des liens inutiles pour les itérations suivantes. Mais cela a pour effet d'accroître le temps de calcul. Une autre approche récente, le "Three Point Clustering" *[SSS98]*, dont la place mémoire requise reste modeste comparée à une méthode classique, nécessite une subdivision très fine pour avoir des effets lumineux de hautes fréquences comme des caustiques. Toutes ces méthodes sont donc coûteuses en temps comme en place mémoire. Dans un autre contexte, les facteurs de forme étendus ont été introduits pour prendre en compte des transferts spéculaires dans le contexte des algorithmes en plusieurs étapes, basés sur la radiosité. Mais souvent ceux-ci sont limités à des objets plans *[WCG87]* ou parfaitement spéculaires *[SP89]*.

De plus, les méthodes déterministes sont encore souvent limitées à des milieux polygonaux où une discrétisation se révèle simple, ceci malgré le développement de nouvelles paramétrisations de la solution *[SSS97, HCA00]*.

3 Les méthodes stochastiques

Les nombreuses limitations des solutions déterministes d'éclairage global ont conduit très tôt les recherches vers les méthodes stochastiques. Dés l'introduction de l'équation du rendu *[Kaj86]*, Kajiya en propose déjà une implémentation. Celles-ci, en effet, n'ont théoriquement aucune limitation sur le type de réflexion et le type de géométrie, et sont peu consommatrices de ressources, mais souffrent d'un problème inhérent à de telles méthodes, le bruit. Depuis, ces méthodes ont évolué dans diverses directions, toujours dans l'espoir d'améliorer l'efficacité de tels algorithmes.

3.1 Méthodes de Monte-Carlo

La thèse de Eric Veach *[Vea97]* constitue une référence sur le sujet. Nous allons ici en introduire simplement les principes généraux.

Dans un cadre déterministe, la résolution numérique d'une intégrale au sens de Lebesgue

$$I = \int_D f(x) d\mu(x) \tag{2.27}$$

se fait à l'aide d'un sommation de la forme

$$I' = \sum_{i=1}^{N} \alpha_i f(x_i) \tag{2.28}$$

où les $(\alpha_i, i = 1..N)$ sont des poids et les $(x_i, i = 1..N)$ des points de discrétisation déterminés. Ces méthodes convergent en $O(N^{-r}), r \geq 1$, si la fonction f possède un nombre suffisant de dérivées continues. Pour une méthode de Simpson, la convergence se fait en $O(N^{-4})$, à condition que la fonction f soit C^4. Lorsque le domaine d'intégration augmente de dimension d, la convergence devient $O(N^{-r/d})$. De telles méthodes supportent difficilement l'augmentation de dimension.

Soient $(X_i, i = 1..N)$ les variables aléatoires associées aux échantillons aléatoires $(x_i, i = 1..N)$, tirés selon une loi de probabilité de densité p sur le domaine D.Alors, l'espérance de l'estimateur

$$I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \tag{2.29}$$

a pour valeur l'intégrale I :

$$\begin{aligned} E[I_N] &= \tfrac{1}{N} E\left[\sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\right] \\ &= \tfrac{1}{N} \sum_{i=1}^{N} E\left[\frac{f(X_i)}{p(X_i)}\right] \\ &= \tfrac{1}{N} \sum_{i=1}^{N} \int_D \frac{f(x)}{p(x)} p(x) d\mu(x) \\ &= I \end{aligned}$$

Lorsque l'espérance de l'estimateur est égale à la valeur de l'intégrale cherchée (si $E[I_N] = \hat{I}$ alors $\hat{I} = I$), il est di sans biais. Le biais définit ainsi l'écart entre la solution souhaitée I et celle de la méthode d'intégration qui permet d'obtenir $\hat{I}$:

$$biais = E[I_N] - I \tag{2.30}$$

L'écart type σ se comporte lui en $N^{-1/2}$, et ceci quelle que soit la dimension du domaine d'intégration. En notant $Y_i = f(X_i)/p(X_i)$, la variance V est :

$$\begin{aligned} V[I_N] &= V\left[\tfrac{1}{N} \sum_{i=1}^{N} Y_i\right] \\ &= \left(\tfrac{1}{N}\right)^2 V\left[\sum_{i=1}^{N} Y_i\right] \\ &= \left(\tfrac{1}{N}\right)^2 \sum_{i=1}^{N} V[Y_i] \\ &= \tfrac{1}{N} V[Y] \end{aligned}$$

Ainsi, $\sigma[I_N] = \sqrt{V[I_N]} = N^{-1/2}\sigma[Y]$. Lorsque la densité p est choisie proportionnelle à f (soit $p = I f$), la variance est nulle :

$$\begin{aligned} V[Y] &= E[Y^2] - E[Y]^2 \\ &= \int_D I^2 p(x) \mu(dx) - I^2 \\ &= I^2 - I^2 \\ &= 0 \end{aligned}$$

Ce résultat nous montre que, plus il y a d'informations connues sur la solution finale, meilleure sera la convergence de la solution.

Ces méthodes sont donc peu sensibles à la dimension du domaine d'intégration. Elles on été pour la première fois utilisées à Los Alamos, dans l'étude de la propagation des neutrons pour la construction de la bombe H.

3.2 Application au problème de l'éclairage

Dans une solution d'éclairage global, la valeur à estimer pour chaque pixel d'une image finale est la somme des contributions lumineuses de l'ensemble des chemins lumineux qui y arrivent : $L(D|S)^*E$ (voir Sec. 1.4). Le nombre de dimensions peut être très élevé ! Dans un cadre plus général qu'une discrétisation, qui suppose communément que le milieu soit diffus, il apparaît alors intéressant d'utiliser de telles méthodes.

En effet, l'espace d'intégration correspond ici à l'ensemble des chemins lumineux possibles. Si $\mathbb{S}$ est l'ensemble des points des éléments de la scène, cet ensemble est $\mathbb{C} = \bigcup_{i=1}^{\infty} \mathbb{S}^i$. En effet, l'équation 2.3 est une formulation récursive. Si nous posons $G'(s,s') = \pi G(s,s')$, elle peut être écrite sous la forme :

$$\begin{aligned} L(s,\omega_o) &= L_P(s,\omega_o) + \textstyle\int_{\mathbb{S}} \rho(s,\omega,\omega_o) G'(s,s') L_P(s',\omega) ds' + \\ &\quad \textstyle\int_{\mathbb{S}} \int_{\mathbb{S}} \rho(s,\omega,\omega_o) G'(s,s') \rho(s',\omega',\omega) G'(s',s'') L_P(s'',\omega') ds'' ds' + .. \\ &= \textstyle\int_{\mathbb{C}} \Pi_{j=0}^{n-1} (\rho(s_j,\omega_{j+1},\omega_j) G'(s_j,s_{j+1})) L_P(s_n,\omega_n) \mu(dc) \end{aligned} \tag{2.31}$$

où n est la longueur du chemin lumineux.

Ainsi, pour construire une tel échantillon, le principe reste le même, que cela soit pour un lancer de rayons *[CPC84, Coo86, Kaj86, Shi90]* à la recherche d'une source de lumière (en partant du récepteur final, l'œil par exemple) ou pour un lancer de particules *[Hec90, Jen96, WHSG97]* ou pour un lancer de rayons à partir des sources lumineuses *[HH84, Arv86, Wat90]*. La méthode en général utilisée est la suivante :

- choix aléatoire des couples (point, direction) qui initialise un chemin lumineux.
- propagation de cet échantillon dans le milieu selon les lois de probabilité définies par les propriétés des matériaux.

Lancer de chemins

Les méthodes de Monte Carlo par lancer de rayons*[Kaj86, Shi90]* sont très populaires de par leur simplicité d'implémentation. Il suffit en fait de choisir aléatoirement un chemin lumineux c (ou rayon) dans l'ensemble des chemins lumineux $\mathbb{C}$, et d'y calculer l'énergie s'y propageant.

$$\begin{cases} E &= \frac{\Pi_{j=0}^{n-1} (\rho(s_j,\omega_{j+1},\omega_j) G'(s_j,s_{j+1})) L_P(s_n,\omega_n)}{p(c)} \\ p(c) &= \Pi_{j=0}^{n} p(\omega_j) \end{cases} \tag{2.32}$$

où $p(c)$ est la densité de probabilité du chemin c, soit

$$E = \frac{1}{p(\omega_0)} \Pi_{j=0}^{n-1} \frac{\rho(s_j,\omega_{j+1},\omega_j) G'(s_j,s_{j+1})}{p(\omega_{j+1})} L_P(s_n,\omega_n) \tag{2.33}$$

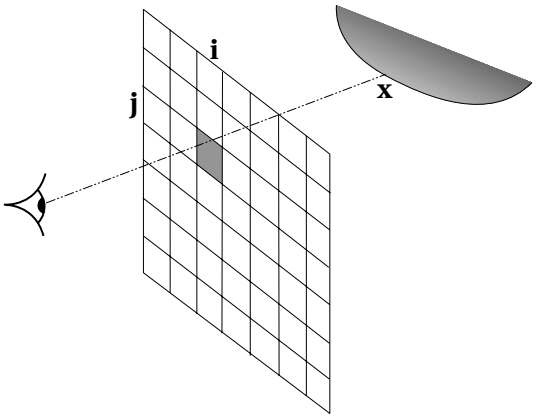


FIG. 2.7 : Émission d'un rayon à partir du point de vue

La valeur en un pixel (i, j) d'une image est alors la moyenne des valeurs des rayons émis à travers ce pixel (cf. Fig. 2.7).

Le principal désavantage d'une telle méthode réside dans le fait qu'il est possible de ne jamais rencontrer une source lumineuse tout au long d'un rayon, la valeur résultante étant alors nulle. La probabilité de rencontrer une source lumineuse étant faible, l'image obtenue est souvent très bruitée (cf. Fig. 2.8-(a)) et la vitesse de convergence est très lente. De plus, ces méthodes ont du mal à capturer des effets lumineux comme des caustiques, pour lesquels un bon échantillonnage de l'énergie provenant des sources lumineuses est essentiel.

Lancer de particules

Dans un lancer de particules, l'émission d'un rayon (assimilé ici à une particule énergétique) se fait à partir des sources lumineuses. Cette particule va alors transporter une partie de l'énergie totale émise par les sources (flux en émission) à travers l'environnement. Les chemins lumineux considérés sont $L(D|S)^*$ *[PM92]*.

Sur une source de lumière S, il est souhaitable que le choix des point s et des directions ω, qui vont initialiser les propagations dans le milieu, soit un bon échantillonnage du flux émis par S : Φ_S^{s+} (cf. Eq. 2.4). C'est à dire que pour un tirage aléatoire de nos échantillons $((s_i, \omega_i), i = 1..N)$, l'espérance de notre estimateur doit être cette valeur de flux. Comme le flux émis est (cf. Eq. 2.4) :

$$\Phi_S^{s+} = \int_S \int_{\Omega^+(s)} L(s,\omega) < \omega \cdot n > d\omega ds \tag{2.34}$$

l'estimateur choisi est alors

$$\begin{cases} I_N &= \sum_{i=1}^{N} \phi_i \\ \phi_i &= \frac{L(s_i,\omega_i) < \omega_i \cdot n_i >}{N p(s_i,\omega_i)} \end{cases} \tag{2.35}$$

La valeur ϕ est l'énergie de la particule émise du point s_i dans la direction ω_i. Cette estimateur est sans biais, puisque $\Phi_S^{s+} = E[I_N]$ (voir section 3.1).

Si la valeur ϕ_i est constante pour chaque particule ($\phi_i = \phi$ avec $\phi = \Phi_S^{s+}/N$), alors quel que soit le nombre choisi d'échantillons, $I_N = \Phi_S^{s+}$. Et dans ce cas :

$$p(s,\omega) = \frac{L(s,\omega) < \omega \cdot n >}{\Phi_S^{s+}} \tag{2.36}$$

Lors des impacts dans l'environnement, les particules sont stockées puis retransmises selon une loi de probabilité $p(\omega)$ permettant de choisir une direction ω de réflexion. Par un même raisonnement que dans les paragraphes précédents, mais cette fois ci sur l'énergie totale réfléchie d'une particule d'énergie ϕ de direction incidente ω en un point s, soit $\phi \int_\Omega \rho(s,\omega,\omega') < \omega' \cdot n > d\omega'$ (cf. Eq. 2.1), l'énergie ϕ' de la particule réfléchie dans une direction ω' devient alors *[Wal98]* :

$$\phi' = \phi \frac{\rho(s,\omega,\omega') < \omega' \cdot n >}{p(\omega')} \tag{2.37}$$

De même, pour que la valeur d'énergie de la particule soit constante le long de sa propagation, il faut que :

$$p(\omega') = \rho(s,\omega,\omega') < \omega' \cdot n > \tag{2.38}$$

Comme $\int_\Omega p(\omega')d\omega' \leq 1$, pour vérifier la conservation de l'énergie, une direction nulle $\omega' = 0$ est rajoutée, correspondant à la probabilité d'arrêt des réflexions (roulette russe) :

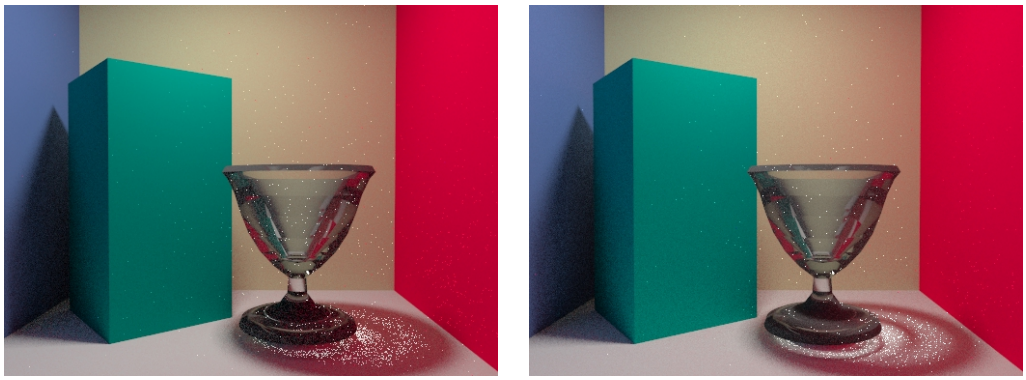
$$p(0) = 1 - \int_\Omega \rho(s,\omega,\omega') < \omega' \cdot n > d\omega' \tag{2.39}$$

L'ensemble des impacts ainsi stocké, représente la fonction d'éclairage incident sur toute les surfaces qu'il faut reconstruire : une carte de l'éclairage *[Hec90]*, une carte de photons *[JC94]*, ou une estimation de densité *[BA97, Wal98, WHSG97]*.

Wann Jensen a introduit le "Photon Map" *[Jen96]*, qui utilise des particules (ou un lancer de photons) pour tous les types de chemins lumineux. Cette méthode peut calculer rapidement des caustiques directes (LS^+D), mais nécessite une étape de reconstruction coûteuse pour calculer les autres chemins lumineux. Malgré l'accélération possible avec l'utilisation du gradient de l'irradiance *[WH92]*, cette étape est trop longue pour devenir interactive.

L'estimation de densité *[WHSG97]* est une approche très similaire. Comme pour le "Photon Map", tous les chemins lumineux sont simulés par un lancer de particules. De plus, le rendu peut être fait en utilisant l'accélération graphique matérielle après une phase de simplification de maillage, ou par un lancer de rayons pour prendre en compte les autres chemins vers l'œil $(L|D)S^+E$.

Cette étape est en général la plus coûteuse en temps de calcul. Wann Jensen *[JC94]* réduit ce coût par une méthode de stockage efficace, qui permet une séparation de



Suivi de chemins simple
100 rayons/pixel - 25mn

Méthode bidirectionnelle
50 rayons/pixel - 31mn

Images générées à l'aide de RenderPark
sur une station SGI Octane2 (processeur MIPS/R12000 400 Mhz)

FIG. 2.8 : Comparaison de deux méthodes stochastiques

la représentation de l'énergie reçue de la géométrie, autorisant ainsi des surfaces complexes comme des surfaces fractales. De plus, le nombre de particules stockées pouvant être élevé, il peut être nécessaire d'avoir une bonne gestion de la mémoire *[WHSG97, SW00]*.

En général, ces méthodes sont plus efficaces qu'une méthode de suivi de chemins simple pour traiter les effets lumineux comme les caustiques, l'éclairage indirect et tous les effets pour lesquels un bon échantillonnage du flux lumineux est requis.

Méthodes bidirectionnelles

Afin de pouvoir simuler ces effets en limitant les ressources nécessaires pour calculer une solution, des méthodes bidirectionnelles ont été introduites *[LW93, VG94, War94, JC94, ZSP98]*. En combinant des rayons à partir du point de vue (chemins $(D|S)^*E$) avec des échantillons provenant des sources lumineuses (chemins $L(D|S)^*$), il est plus facile de capturer les effets lumineux comme des caustiques (cf. Fig. 2.8), des phénomènes dus à un éclairage indirect, tout en gardant la simplicité d'un lancer de rayons.

Méthodes de Metropolis

Les méthodes de Metropolis *[VG97]* sont d'autres extensions des méthodes probabilistes pures. Elles sont basées sur la perturbation de chemins lumineux. Ces perturbations vont leur permettre théoriquement de bien mieux s'adapter à des configurations difficiles. La principale difficulté de ces méthodes réside dans le choix d'une telle fonction de perturbation adéquate, choix qui se révèle souvent crucial. Elles restent parmi les méthodes probabilistes les plus difficiles à mettre en oeuvre.

3.3 Les méthodes hybrides

Dans la recherche d'algorithmes efficaces pour le calcul de l'éclairage global, de nombreuses solutions hybrides ont été proposées dans le but d'accroître les capacités de chacun des algorithmes.

Méthodes en deux étapes

Une des approches les plus simples consiste à calculer une solution par éléments finis, indépendante du point de vue en guise de première étape, et d'y rajouter par lancer de rayons les composantes dépendantes du point de vue pour le calcul de l'image finale *[WCG87, WEH89, SP89]*.

En fait, dans une première étape, la solution calculée représente les chemins lumineux LD^*, voire $L(D|DS^1D|...|DS^nD)^*$ pour les méthodes plus directionnelles (en général $n = 1$).

Méthodes stochastiques sur maillage

Il y a dans ces approches *[FP93, SPP95, TWFP97, NNB97, BNN+98, Bek99]*, la volonté de garder ce qui fait la force des algorithmes déterministes, c'est à dire le fait de pouvoir projeter la solution dans une base de fonctions définie sur un maillage. Le calcul est indépendant du point de vue, et la discrétisation est basée sur un maillage.

Pour réduire la complexité de l'intégration sur un domaine qui reste de grande dimension, le choix est fait de calculer les coefficients de discrétisation par des méthodes stochastiques. Des rayons propagés dans toute la scène vont représenter l'ensemble des échanges entre les surfaces. Ces algorithmes ont pour principal défaut leur difficulté à capturer les effets de hautes fréquences d'une solution, car l'approche reste très globale. Cependant, leur principale avantage est d'être plus ou moins indépendant de la complexité de la scène.

Méthodes en plusieurs étapes

Une autre approche consiste à intégrer toutes les diverses méthodes au sein d'un même algorithme en plusieurs étapes, chaque étape ayant en charge le calcul d'une certaine partie de la solution en employant la méthode la plus efficaces pour ceci *[Shi90, Shi91, Neu95, SW99]*.

Le meilleur exemple, et précurseur de nos travaux, est l'algorithme proposé par Chen et al. en 1991 *[CRMT91]*. Il est principalement basé sur l'intégration d'une méthode de radiosité progressive, avec des lancers de rayons à partir de sources lumineuses (pour les effets comme les caustiques) et à partir du point de vue (pour les reflets). La radiosité progressive sert à calculer la solution d'un éclairage diffus, solution qui sera corrigée et améliorée par les lancers de rayons successifs, permettant ainsi l'obtention d'un plus grand nombre de détails que le permettrait une solution de

radiosité simple, comme des ombres plus précises, des caustiques, des effets directionnels.

Cependant, l'intégration reste très complexe, et la prise en considération des différents méthodes nécessite de nombreuses opérations sur les valeurs d'énergie lumineuse (additions, soustractions, corrections). De plus, dans le cas d'un éclairage majoritairement indirect, une reclassification des sources secondaires est nécessaire, augmentant encore la complexité du calcul.

3.4 État des approches stochastiques

Parmi toutes les solutions d'éclairage global, les méthodes stochastiques ont montré leur capacité à produire des images de très haute qualité, ceci grâce à leur capacité à bien capturer les hautes fréquences (comme les caustiques, les ombres dures, et les effets hautement directionnels) et leur relative indépendance à la complexité des scènes. Mais leur convergence restent assez longue, notamment à cause du phénomène de bruit.

Par des méthodes de reconstruction plus poussées (estimations de densité, utilisation d'un maillage), ce phénomène est réduit mais au détriment de certaines hautes fréquences, et surtout, un biais est introduit. Les méthodes hybrides, par la séparation des différentes composantes de la lumière réfléchie, tendent à tirer avantage des différentes méthodes, mais les mises en commun effectuées sont soit une simple superposition (méthodes en deux étapes) limitant les effets possibles, soit une intégration complexe rendant l'algorithme difficilement contrôlable.

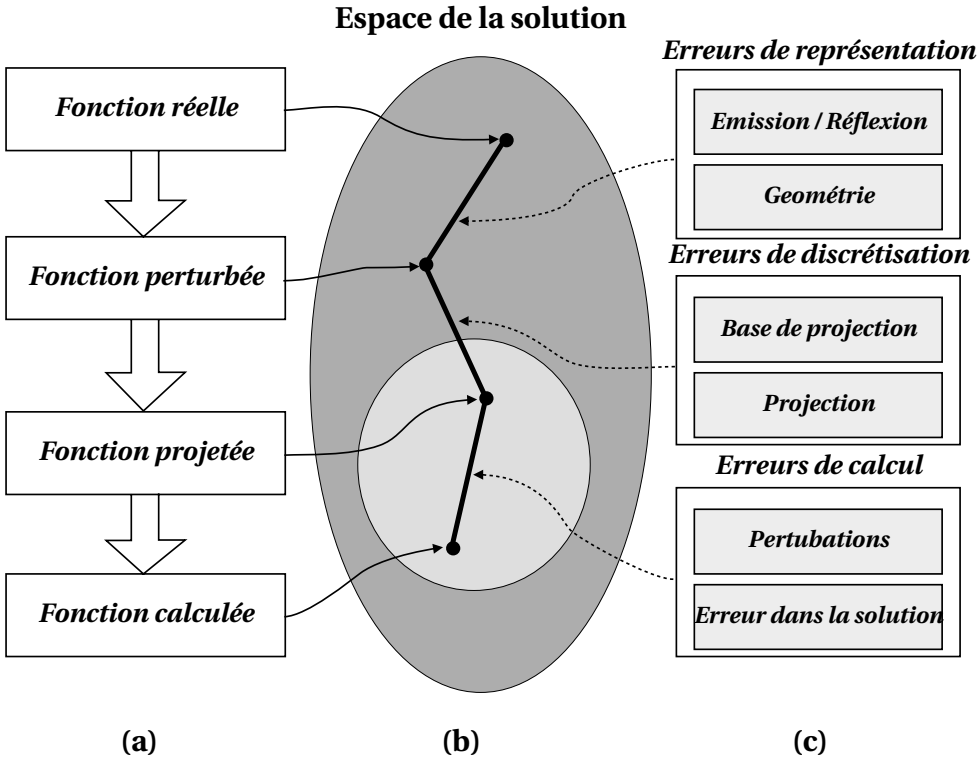
4 Réflexion sur le contrôle de la qualité pour un éclairage global

A travers ce tour d'horizon très rapide, nous pouvons constater qu'au cours de ces vingt dernières années, les méthodes d'éclairage global ont proliféré, chaque innovation répondant à un nouveau problème spécifique, mais sans forcément prendre en compte les avantages des solutions déjà proposées. De toute cette expérience accumulée, il est possible de déterminer quel sont les critères qui permettraient un contrôle de la qualité. Pour cela, il est possible d'aborder le problème de la qualité sous deux angles : l'un étant une étude de l'erreur commise lors de la simulation, l'autre une approche plus perceptuelle.

4.1 Contrôle de l'erreur commise

Arvo et al *[ATS94]* ont présenté en 1994 une étude des différentes étapes d'une simulation d'éclairage en précisant là où peuvent intervenir des erreurs et approximations (cf. Fig. 2.9). Cette démarche est essentielle à une bonne compréhension des

problèmes rencontrés pour contrôler le résultat final d'une simulation d'éclairage global.



(a) étapes de la simulation - (b) espace des solutions - (c) erreurs et approximations

FIG. 2.9 : Les étapes de la simulation d'éclairage (adapté de *[ATS94]*)

.

Modélisation réaliste de la scène

Dans le cadre d'une simulation, les premières erreurs viennent donc de la modélisation de la scène. Il est ainsi nécessaire de posséder un modèle le plus "réaliste" possible. Dans nos travaux, nous supposerons que cette tâche est accomplie. Le but sera de partir d'une scène donnée, et d'obtenir le meilleur résultat possible pour son éclairage. Dans ce cadre, un polygone n'est pas la seule primitive qu'il faut considérer. Faire une telle supposition augmenterait considérablement les erreurs faites. Ainsi une modélisation de plus haut niveau, comme des sphères, des cylindres, permettrait de réduire les erreurs dues à la représentation interne ou à la discrétisation.

De même, il est nécessaire d'avoir une bonne modélisation des propriétés de réflexion, afin de ne pas se limiter à des modèles courants, généralement les modèles de Phong *[Pho75]*. Une représentation d'une propriété lumineuse doit donc ainsi être suffisamment abstraite afin de toujours permettre le traitement de plus de matériaux possibles.

Rendre toutes les réflexions

Une des premières nécessités pour avoir un éclairage réaliste de haute qualité, c'est de pouvoir rendre tous les types d'échanges lumineux, c'est à dire, de générer tous les chemins $L(D|S)^*E$ (cf. Sec. 1.4). Le grand apport des algorithmes d'éclairage global a été le traitement de l'éclairage indirect, permettant la génération d'images de grande qualité.

Ainsi, le calcul de tous les phénomènes lumineux possibles réduit les erreurs faites dans le passage d'un modèle donné vers notre base de représentation. C'est parce qu'il existe une telle approximations à la base que les algorithmes de radiosité ont du mal à évoluer vers des algorithmes plus généraux.

Bonne représentation de toute l'échelle de variation

Dans une image finale, l'œil est capable de discerner une grande variation de détails. Les phénomènes de basses fréquences, comme les réflexions diffuses, permettent de définir une ambiance générale. C'est cet apport qui a fait la force des algorithmes de radiosité.

Cependant, l'œil est aussi capable de voir des phénomènes de très hautes fréquences, comme les caustiques et les ombres. Ce sont ces détails qui ont fait la force des algorithmes stochastiques.

Un algorithme efficace devrait donc être capable de générer ce large éventail de détails, et donc de le calculer et de le représenter efficacement

Gestion de la complexité

A travers toutes ces remarques, il apparaît nécessaire de gérer la complexité intrinsèque des modèles "réalistes", contenant des milliers, voire des millions d'objets comme des polygones. C'est le grand enjeu des algorithmes d'éclairage global. La formulation est assez bien posée, le problème est d'évaluer cette équation sur des modèles de grande complexité, que cela soit sur la géométrie ou les types de matériaux.

De plus, il peut être intéressant, pour l'utilisateur, d'avoir un retour rapide de ce que peut devenir une solution finale. Ce premier résultat peut guider dans le contrôle efficace de la qualité finale d'une solution.

4.2 Contrôle de la qualité visuelle

Il est possible d'aborder la qualité sous une angle plus perceptuel. En effet, entre une image de radiosité sans aucun lissage, avec une base constante par élément, et cette même solution sur laquelle un lissage a été effectué, la seconde image est en générale considérée comme de plus haute qualité, même si elles sont équivalentes numériquement. En effet, notre oeil est très sensible aux discontinuités visuelles, et lorsqu'il en existe une qui ne semble pas naturelle dans une image, il y a une forte dépréciation de la qualité.

De même, le bruit inhérent aux méthodes probabiliste diminue la qualité des images produites, même si tous les effets voulus sont perceptibles. C'est pourquoi il est souvent nécessaire de faire appel à des méthodes de reconstruction *[Hec90, JC94, WHSG97]* ou de filtrage *[McC99]* afin de recréer une continuité visuelle perturbée par le bruit.

Dans cette approche, des méthodes moins physiques, mais plus intuitives, peuvent permettre d'améliorer la qualité d'une solution : approximation des ombres, approximation des effets directionnels.

4.3 Automatisation

Il est possible à l'heure actuelle de simuler l'ensemble des effets lumineux qui rendent un environnement virtuelle réaliste. La plupart des problèmes ont été identifiés, et une ou plusieurs solutions ont été proposées.

La difficulté aujourd'hui réside dans la combinaison de tous ces effets et dans le contrôle du résultat de cette combinaison. En effet, un utilisateur se trouve devant un large éventail de solutions possibles, chacune ayant prouvé son efficacité dans des configurations précises. Le choix n'est pas toujours trivial, et les superbes images qu'il est possible de voir sont le résultat d'un très grand labeur de la part des utilisateurs.

Il devient donc nécessaire de proposer une méthode plus généraliste, automatisant cette intégration pour un meilleur contrôle sur la qualité finale.

3

Plateforme d'expérimentation

> Ne vous laissez pas abuser pas les surfaces, en profondeur tout est loi.
>
> Rainer Maria Rilke
> *Lettre à un jeune poète*

Notre étude menée sur le contrôle automatique de qualité pour une solution d'éclairage global a nécessité la mise en place d'une plateforme d'expérimentation : G.I.S. (pour "Global Illumination System"). Nous présenterons donc ici à la fois l'architecture logicielle développée dans le cadre de cette expérimentation, mais aussi les solutions et les choix retenus dans la conception d'un algorithme d'éclairage global, principalement basé sur une méthode de radiosité hiérarchique. Cette conception a permis entre autre une première approche des problèmes soulevés dans le chapitre précédent. Les objectifs qui nous ont guidés sont ceux de la rapidité d'exécution pour une part, de même que la facilité d'implémentation d'extensions.

Ce chapitre est composé de quatre sections abordant les problèmes du traitement hiérarchique essentiel pour les scènes complexes, puis les problèmes de lissage et enfin la procédure de raffinement choisie. Mais en premier lieu, nous présenterons les choix effectués pour l'architecture de cette plateforme.

1 Architecture logicielle : G.I.S.

Par la mise en place de l'architecture de la plateforme d'expérimentation G.I.S.[1], nous voulons déjà aborder certains aspects du contrôle de qualité. Une structure modulaire[2] a ainsi été choisie, permettant de tester diverses approches.

1.1 Modules généraux

Le premier choix effectué, est la possibilité d'avoir une géométrie complètement indépendante. Ce chapitre est composé de quatre sections abordant les problèmes du traitement hiérarchique essentiel pour les scènes complexes, puis les problemes de lissage simple et enfin la procédure de raffinement choisie. Mais en premier lieu, nous présenterons les choix effectués pour l'architecture de cette plateforme, comme indiqué dans la figure Fig. 3.1, mais qui répond à certaines spécifications. Cette indépendance permet de traiter avec précision un objet géométrique défini sans pour autant passer par une représentation polygonale. Un objet géométrique est ainsi chargé de gérer tout ce qui est relatif aux calculs d'intersections, aux mesures de volume et d'aire, à l'échantillonnage, et à la subdivision. Il peut de plus être étendu par les calculs exacts de facteurs de forme. Nous avons ainsi développé un module de base, contenant les objets géométriques de type boîte englobante, polygone généralisé et spécialisé (triangle, quadrilatère), et enfin sphère, cône, cylindre, anneau, ceci pour avoir une palette de géométrie assez variée. La paramétrisation de ces objets est présentée dans l'annexe A.

Dans le même esprit, les propriétés matérielles sont relativement indépendantes, mis à part de la géométrie (cf. Fig. 3.1). En effet, un matériau doit pouvoir interroger la géométrie afin de déterminer la position, la normale, l'orientation et autres, notamment pour les textures ou les matériaux anisotropiques. Cette séparation est équivalente à celles effectuées pour les modèles d'éclairage ("shaders") des logiciels de modélisation existants[3] ou des systèmes de rendu[4]. Cela permet d'introduire facilement de nouveaux matériaux et ainsi, d'améliorer les possibilités de rendu. Cela a été pensé pour expérimenter l'utilisation de propriétés matérielles diversifiées. Nous avons développé un module comportant des matériaux diffus, des réflexions de type Phong et miroir, des réfractions de ces deux types, et enfin des matériaux diffus texturés. Nous avons ainsi une bonne variation de la directionnalité des matériaux, et de leur répartition spatiale.

Nous avons par la suite développé un module chargé de gérer la représentation hiérarchique de la scène, afin de gérer au mieux la complexité. C'est aussi à ce niveau

[1] Une documentation plus complète se trouve en ligne à l'adresse suivante
`http ://www-imagis.imag.fr/~Xavier.Granier/GIS/html/index.fr.html`

[2] En pratique, pour chaque module, une librairie a été développée. Leurs description se retrouve dans l'annexe C.

[3] comme Maya ©Alias-Wavefront

[4] comme pour RenderMan ©Pixar

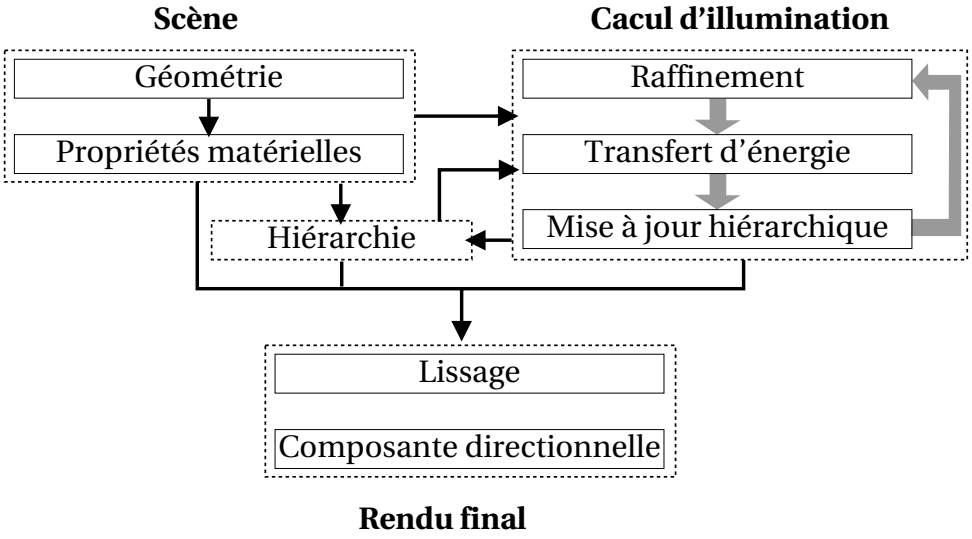


FIG. 3.1 : Architecture du système d'éclairage global

que se fait l'association entre la géométrie et les propriétés de réflexion (cf. Fig. 3.1). C'est cette structure que va construire notre module d'entrée de données. Ce module est de plus chargé de simplifier la géométrie (comme un maillage) lorsque cela est possible, mais aussi de convertir les informations de matériau en des propriétés matérielles physiquement réalistes.

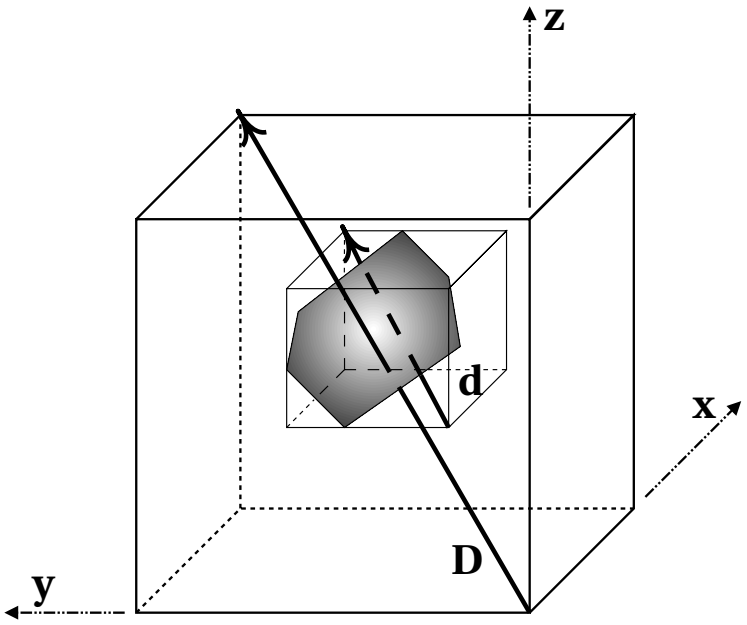
Afin d'augmenter la hiérarchisation des modèles traités, un module de regroupement a aussi été introduit (cf. section Sec. 2.1). De même, pour les besoins de tout algorithme d'éclairage, des module de lancer de rayons et de calcul de visibilité (cf. Sec. 3.2) ont été mis en place.

1.2 Module de radiosité hiérarchique

Jusqu'à présent, il n'y a aucune spécialisation pour une solution d'éclairage global donnée. Cette généralité des modules permettra par la suite de tester différentes méthodes de simulation d'éclairage global. Dans le cadre de méthodes déterministes, des méthodes classiques de maillage peuvent être utilisées, mais aussi des techniques comme des textures *[Hec90, MK94, BGB97]*, des maillages indépendants *[HCA00]*, ou tout autre représentation alternative de la solution d'éclairage.

Le traitement de l'éclairage, se fait ainsi à l'aide d'une structure à part. Comme dans la plupart des cas, le processus est itératif, hérité des algorithmes de radiosité hiérarchique *[HSA91]*. Chaque itération est en général composée d'une phase de positionnement des échanges énergétiques, souvent appelée raffinement, suivie d'une étape de transfert d'énergie, et pour finir d'un étape de mise à jour de la solution d'éclairage.

C'est à partir des informations contenues dans ce module qu'il est possible de faire une étape de reconstruction pour l'affichage (cf. Sec. 4) et de rendu final.



$d = (d_x, d_y, d_z)$ et $D = (D_x, D_y, D_z)$

FIG. 3.2 : Géométrie du test de regroupement

2 Traitement hiérarchique

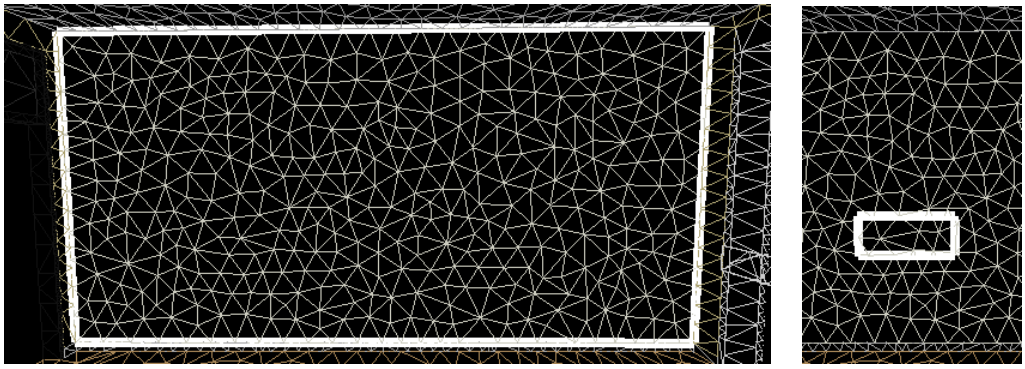
Avec l'accroissement constant de la complexité des scènes dû à la volonté d'obtenir des modèles réalistes, il est absolument nécessaire d'utiliser une structure hiérarchique. Comme pour le cas des algorithmes de radiosité, l'emploi d'une telle organisation des environnements est vite apparu indispensable pour compenser un coût de calcul implicite en $O(N^2)$ pour toute simulation des échanges globaux.

Le premier problème consiste à factoriser les calculs en regroupant des objets ensembles. Nous décrirons dans un premier temps notre méthode de regroupement, puis comment utiliser les groupes afin d'améliorer leur traitement et de retrouver une information de direction.

Il faut aussi être capable de traiter hiérarchiquement des objets de plus haut niveau, comme des sphères, cylindres, polygones de plus de quatre sommets. Cette hiérarchisation permettra de traiter des scènes plus complexes dans les types de géométrie utilisables.

2.1 Regroupements

Depuis l'introduction des regroupements *[SAG94, SDS95]* dans les algorithmes de radiosité hiérarchique, de nombreuses méthodes ont été proposées *[SAG94, SDS95,*



(a) (b)

(a) Méthode originale (b) Notre solution

FIG. 3.3 : Première amélioration du regroupement

CLSS97]. L'étude de Hasenfratz et al. *[HDSD99]* a montré la difficulté de choisir parmi elles une méthode de regroupement générale, qui s'adapte bien à la plupart des cas.

Par sa simplicité et sa rapidité sur les scènes complexes, nous avons décidé de baser notre approche sur une hiérarchie de boîtes englobantes et sur l'algorithme de Christensen et al. *[CLSS97]*. Il ne comporte aucun prérequis vis à vis de la modélisation de la scène et permet de regrouper des objets déjà organisé hiérarchiquement. Considérons la figure Fig. 3.2. Dans l'algorithme originel, l'objet peut être descendu dans une des octants de la boîte englobante si sa propre boîte englobante vérifie la condition Q suivante :

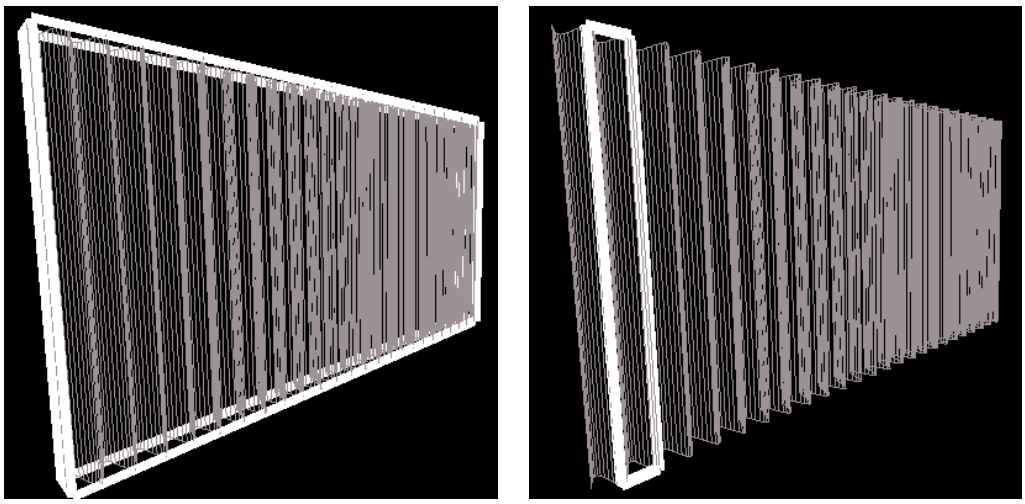
$$Q \;=\; (d_x < 0.5D_x) \wedge (d_y < 0.5D_y) \wedge (d_z < 0.5D_z) \tag{3.1}$$

Les symboles $d_{x,y,z}$ et $D_{x,y,z}$ sont définies dans la figure Fig. 3.2. L'octant choisi est celui contenant le centre de gravité de l'objet. Si le test n'est pas vérifié, l'objet reste au niveau courant. Une fois que tous les objets sont traités, la boîte englobante est reconstruite pour chacun des octants à partir des objets contenus. La procédure de regroupement peut alors se répéter sur chacun de ces nouveaux groupes. Cette méthode connaît une limitation dans le cas où les objets sont planaires et alignées avec les axes. Elle n'est pas capable de les regrouper (cf. Fig. 3.3-(a)).

Nous proposons l'extension suivante du premier critère :

$$\begin{cases} Q^1 &= (d_x \leq 0.5D_x) \wedge (d_y \leq 0.5D_y) \wedge (d_z \leq 0.5D_z) \\ Q^2 &= (d_x \neq D_x) \vee (d_y \neq D_y) \vee (d_z \neq D_z) \\ Q' &= Q^1 \wedge Q^2 \end{cases} \tag{3.2}$$

Dans ce nouveau critère Q', le test Q^1 est une simple modification du test Q pour tenir compte de ce cas planaire, et le test Q^2 pour éviter une récursion infinie possible dans le cas où la boîte englobante est de volume nul (boîte assimilable à un plan). Un résultat



(a) (b)

(a) critère strict (b) critère large

FIG. 3.4 : Seconde amélioration du regroupement

est visible dans la figure Fig. 3.3-(b). Mais dans certain cas, ce test reste insuffisant et crée des groupes contenant encore trop d'objets, notamment lorsque tous les objets sont de dimension équivalente (cf. Fig. 3.4-(a)). En effet, dans cette configuration, il n'est pas possible de trouver un octant pouvant contenir ces polygônes, car ceux-ci sont de même dimension que la boîte englobante.

Nous traitons ce cas à l'aide d'une autre étape de regroupement, basée sur trois critères successifs plus larges, comme le critère Q'_x, selon l'axe x, suivant :

$$\begin{cases} Q^1_x &= (d_x \leq D_x) \wedge (d_y < D_y) \wedge (d_z < D_z) \\ Q'_x &= Q^1_x \wedge Q^2 \end{cases} \tag{3.3}$$

Les critères Q'_y, selon l'axe y, et Q'_z, selon l'axe z, sont définis de même. Le résultat obtenu est visible sur la figure Fig. 3.4-(b).

L'algorithme de regroupement est maintenant constitué de quatre tests pour chaque niveau hiérarchique. Tous les objets (surface ou groupe) du niveau courant sont tout d'abord regroupés suivant le critère Q'. Une fois ces groupes construits en tant que fils du niveau courant, un nouveau regroupement est fait à ce niveau selon le critère Q'_x, puis Q'_y, et enfin Q'_z pour tenir compte des cas extrêmes.

De plus, par cette méthode, il est possible de créer des groupes plans, comme dans la figure Fig. 3.3. Dans ce cas là, ils peuvent être considérés comme des hiérarchies de faces, de même que dans *[WHG99]*.

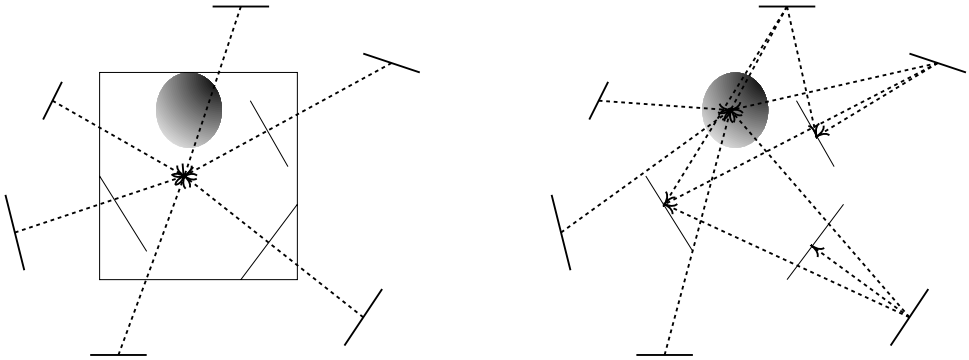


FIG. 3.5 : Descente des liens sur les objets

2.2 Calculs sur des groupes

L'utilisation des groupes a permis le traitement de scènes de plus en plus complexes par les algorithmes de radiosité hiérarchique. L'inconvénient principal réside dans la perte de toute information de direction. Il est possible pour corriger cela de garder une représentation directionnelle de l'énergie émise par le groupe *[ICG86, SDS95, SSG+99]*, mais comme nous l'avons remarqué dans l'état de l'art (cf. Chap. 2-Sec. 2.6), ces méthodes souffrent d'un grand coût mémoire.

Notre stratégie de traitement des groupes va séparer le traitement en deux aspects, la directionnalité de l'émission et celle de la réception. Lorsqu'un groupe est émetteur dans un échange, son ouverture va être favorisée dans le processus de raffinement, ceci dans le cas où il émet une forte énergie. Cela permet de réduire les erreurs commises dans ce type de configuration. Cela sera plus détaillé lors de la présentation de la stratégie de raffinement (cf. Sec.3).

Dans le cas du récepteur, il est possible de tenir compte plus simplement de l'orientation des surfaces. Dans le même esprit que *[SSG+00]*, les liens sont descendus sur les surfaces (cf. Fig. 3.5), et la valeur d'irradiance transportée par ces dernières est corrigée par le cosinus de l'angle entre la direction principale de l'échange et la normale de la surface. C'est une répartition du flux qui se fait, permettant que les surfaces qui ne voient pas l'émetteur ne soient pas éclairées.

2.3 Traitement des objets

Pour pouvoir prendre en compte des objets comme des sphères, des cônes, polygones quelconques, nous avons une représentation par objet dans notre système de radiosité (cf. Annexe. A). Celle-ci est similaire, dans l'approche, à *[SSS97, Sch97]*, dans le sens où, pour la simulation, ces objets ne sont pas présubdivisés, mais cela peut être effectué à la demande. Le nombre de primitives d'une scène est donc réduit. De plus, les calculs d'intersections se font avec les objets initiaux, (donc non subdivisés),

réduisant le temps de calcul et améliorant la qualité des ombres et des caustiques. L'indépendance de notre module de géométrie permet de traiter ces problèmes efficacement.

Il faut cependant certains prérequis sur les fonctionnalité de ces objets géométriques, comme la possibilité de les subdiviser de façon hiérarchique, la possibilité de calculer la normale en un point et de trouver une paramétrisation implicite de textures (cf. Chap. 5).

3 Raffinement

L'objectif qui a guidé la mise en place de l'oracle de raffinement (cf. Chap. 2-Sec. 2.4), c'est dans un premiers temps la simplicité de mise en oeuvre, mais aussi la rapidité de traitement d'une solution tout en étant conservatif au cours des itérations. Une approximation peut être faite à condition de garder l'information comme quoi le calcul existant n'est pas complètement fiable. Il ne doit, par exemple, pas être possible de déclarer deux objets comme étant complètement visibles par un traitement qui ne serait pas exact.

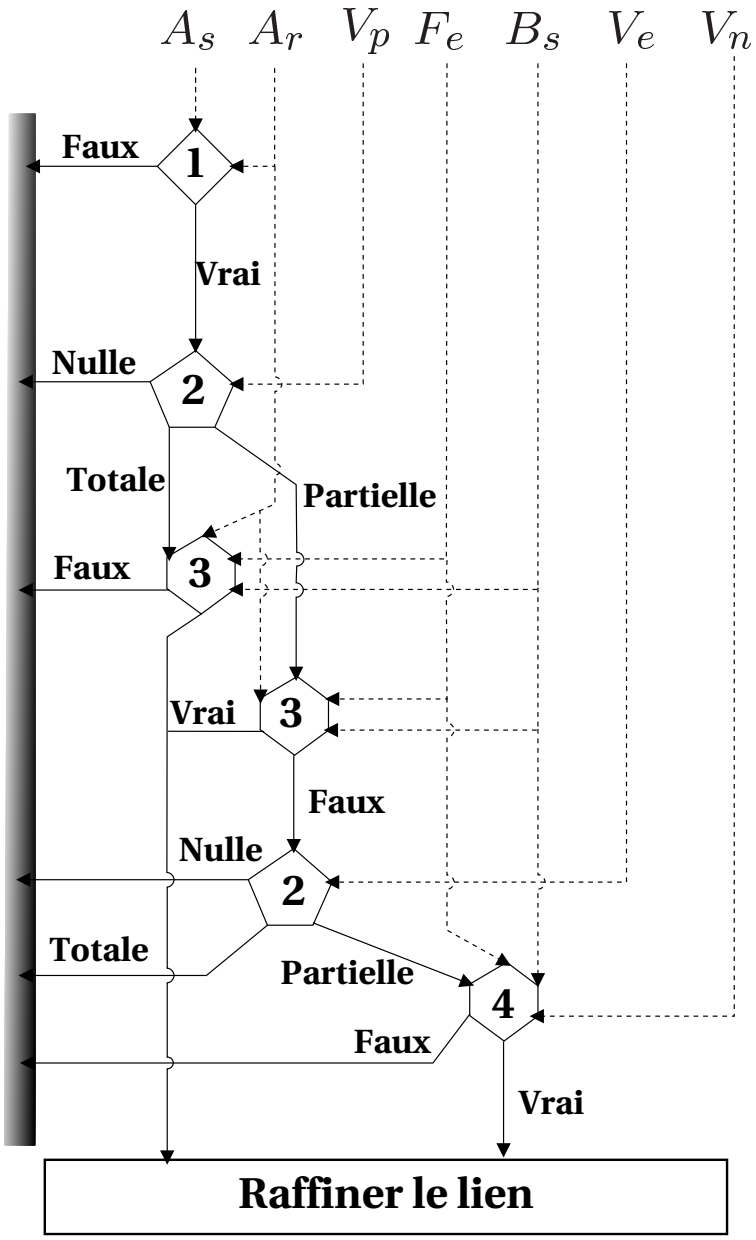
3.1 Oracle de raffinement

L'oracle de raffinement est un critère énergétique *BFA [DSSD97]*, soit l'irradiance multiplié par l'aire du récepteur. Il est supposé que l'erreur commise par la création d'un lien est proportionnelle à l'énergie qui le traverse. Ce critère est augmenté d'un facteur F_V lorsqu'il y a une incertitude sur la visibilité[1]. Il devient alors $BFAF_V$. Le graphe de décision de l'oracle *[TS01]* est représenté dans la figure (cf. Fig. 3.6).

Le choix de l'élément à subdiviser fait partie intégrante du choix de raffinement. Dans notre cas, nous avons choisi un test simple, qui tente d'équilibrer la taille du récepteur et de l'émetteur, tout en essayant de réduire l'erreur produite par la perte de la directionnalité dans les groupes émetteurs. En effet, si l'émetteur est un groupe, le choix de subdivision se porte en priorité sur lui. Sinon, si un seul des deux éléments est de taille suffisante, il sera bien sûr choisi. Si enfin les deux peuvent être subdivisés, l'émetteur est choisi s'il est au moins quatre fois plus grand que le récepteur, ce qui donne une priorité au récepteur.

Il faut remarquer que des valeurs plus précises du facteur de forme et de la visibilité ne sont calculées qu'à la création du lien. L'oracle ne travaille que sur des valeurs estimées. Cela permet d'avoir un raffinement rapide. L'itération suivante se basera sur ces nouvelles valeurs pour repositionner le lien si l'erreur commise a été trop grande.

[1] F_V est dans la majorité des cas fixé à 50.



A_r = aire du récepteur, A_s = aire de l'émetteur, V_p = visibilité du lien père (Nulle, Totale, Partielle), F_e = facteur de forme estimé, B_s = radiosité de l'émetteur, V_e = estimation de la visibilité (Nulle, Totale, Partielle), V_n = visibilité calculée par n rayons ($V_n \in]0,1[$)
Condition 1 : $A_r \geq \varepsilon_A$ ou $A_s \geq \varepsilon_A$ où s est un groupe.
Condition 2 : valeur de la visibilité (Nulle, Totale ou Partielle)
Condition 3 : $B_s \times F_e \times A_r \geq \varepsilon$
Condition 4 : $B_s \times F_e \times A_r \times F_V \times (1 - V_n) \geq \varepsilon$, avec F_V, le facteur de visibilité

FIG. 3.6 : Oracle de raffinement

3.2 Calcul de visibilité et du facteur de forme

Nous utilisons le "shaft culling" *[HW94]* pour accélérer le calcul de visibilité. Comme dans *[Tur01]*, pour un lien donné, une liste d'obstacles possibles est maintenue. Les groupes dont la boîte englobante intersecte le volume d'interaction (ou "shaft" en anglais) sont ouverts. Les éléments qui sont complètement à l'intérieur sont gardés, tandis que ceux qui sont à l'extérieur sont enlevés. La visibilité totale correspond à une liste vide. Lorsqu'un lien est subdivisé, la liste ainsi créée sert de base à la construction de celles des liens fils. Cette construction récursive permet de construire des listes grossières pour des échanges à haut niveau, et d'affiner les résultats au fur et à mesure de la descente dans la hiérarchie.

Pour limiter le coût de la création de cette liste d'obstacles potentiels, les groupes de la hiérarchie ne sont pas systématiquement ouverts, de même que dans *[HW94]*. La stratégie que nous avons adoptée est la suivante :

- Si la boîte englobante du groupe contient un des deux éléments pour lesquels le test de visibilité est effectué, elle est toujours ouverte.
- Sinon, dans le cas où elle est en interaction avec le volume d'interaction, elle n'est ouverte que sur un niveau.
- Enfin, la récursion s'arrête dès qu'un objet se trouve à l'intérieur du volume d'interaction.

L'estimation de la visibilité se fait sur cette liste. Une liste vide correspond bien sûr à une visibilité totale. Sinon, la visibilité est déclarée comme partielle. S'il est demandé de tester la visibilité nulle, cette liste est parcourue, les boîtes de taille suffisante par rapport au volume d'interaction sont ouvertes, et la visibilité n'est testée que sur les objets convexes de taille suffisante[1].

Le calcul de la visibilité se fait, dans le cas partiel, par un échantillonnage des éléments en interaction *[HSA91]*, en répartissant le nombre d'échantillons par rapport aux aires projetées, afin de les utiliser au mieux. Il est recommandé d'éviter de prendre des échantillons sur les bords d'un objet. En effet, dans le cas d'un maillage, ces bords sont communs à deux éléments voisins. Ainsi, un rayon passant par le bord d'un objet peut être intersecté par son voisin, faussant le calcul de visibilité.

Pour le facteur de forme, l'estimation se fait sur un seul échantillon (calcul point à point). Un calcul plus précis est effectué à la création du lien. Ce calcul se fait dans le cas général par un échantillonnage des objets en interaction, et une moyenne des facteurs de forme de point à disque, mais lorsque il existe un calcul exact d'un point vers l'objet considéré *[How82]*, une moyenne de points vers objet est faite. Il est recommandé de tenir compte des bords (au contraire de la visibilité) pour avoir un calcul conservatif sur les objets polygonaux. En effet, dans ce cas, si le calcul de facteur est égal à zéro sur tous les échantillons, alors il est certain qu'il n'y a aucune interaction entre les objets polygonaux convexes. Si seul des échantillons internes sont utilisés,

[1] L'approche est dans le même esprit que celle de *[Tur01]*.

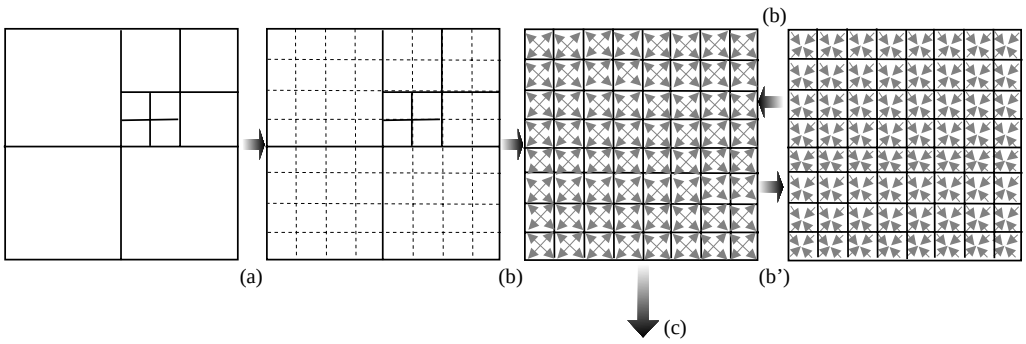


FIG. 3.7 : Construction de la grille

deux éléments hiérarchiques peuvent être déclarés comme n'étant pas en interaction, et cela de manière erronée.

4 Lissage

Nous présenterons ici divers techniques de lissage adaptées à notre représentation par objets. La sortie de cette étape est un ensemble de polygônes, utilisé pour l'affichage interactif ou pour le rendu d'une image. La première approche, simple, s'adapte très bien à l'affichage des surfaces courbes, mais est grande consommatrice de ressources. Les suivantes n'utilisent que la subdivision existante mais permettent ainsi de limiter les ressources nécessaires.

4.1 Utilisation d'une grille

Pour un quadrilatère par exemple (cf. Fig. 3.7), une grille régulière est crée, et chacun de ses polygones a la taille du plus petit objet de la subdivision de ce quadrilatère. Cette approche est adaptable à tout type d'objet. Il est de plus possible de définir une profondeur minimale pour forcer la subdivision et ainsi avoir une bonne représentation des objets courbes.

Le lissage se fait itérativement sur cette grille. Pour chaque feuille de la hiérarchie, sa valeur de radiosité constante est affectée à chacun des polygones de la grille qu'elle contient. Sur chacun des points de la grille, la moyenne des valeurs des polygones adjacents est alors calculée (cf. Fig.3.7-(b)). Si un lissage plus fort est souhaité, il est possible d'affecter itérativement au polygone la moyenne des valeurs de ses sommets (cf. Fig. 3.7-(b')), et de recalculer la valeur des sommets comme la moyenne des valeurs des polygones adjacents. Celle solution permet de forcer le lissage à travers plusieurs polygones et donc d'augmenter l'étalement des valeurs. L'influence du nombre d'itérations est bien visible dans la figure Fig. 3.8. Une fois la solution calculée, l'utilisateur peut modifier le nombre d'itérations et ainsi contrôler la qualité qu'il souhaite.

0 2 10

Remarquez la disparition progressive des détails et notamment des ombres.

FIG. 3.8 : Variation du nombres d'itérations de lissage sur une grille

4.2 Structure de point

Le gros défaut de la méthode précédente est la création d'une grande quantité de polygones pour l'affichage. Pour des objets plans, il est préférable de récupérer les points qui sont sur le contour, même ceux dus à la subdivision des polygones voisins, ceci afin d'éviter des discontinuités de lissage. La procédure de lissage se fait alors sur chaque point ainsi trouvé, en calculant la moyenne des valeurs des polygones adjacents.

Cette solution est moins consommatrice de polygones, car elle s'adapte à la subdivision existante (au contraire de la grille qui crée toujours une subdivision maximum). Mais elle s'adapte mal à la bonne représentation des courbures des objets autres que polygonaux, pour lesquels la grille représente la meilleure alternative.

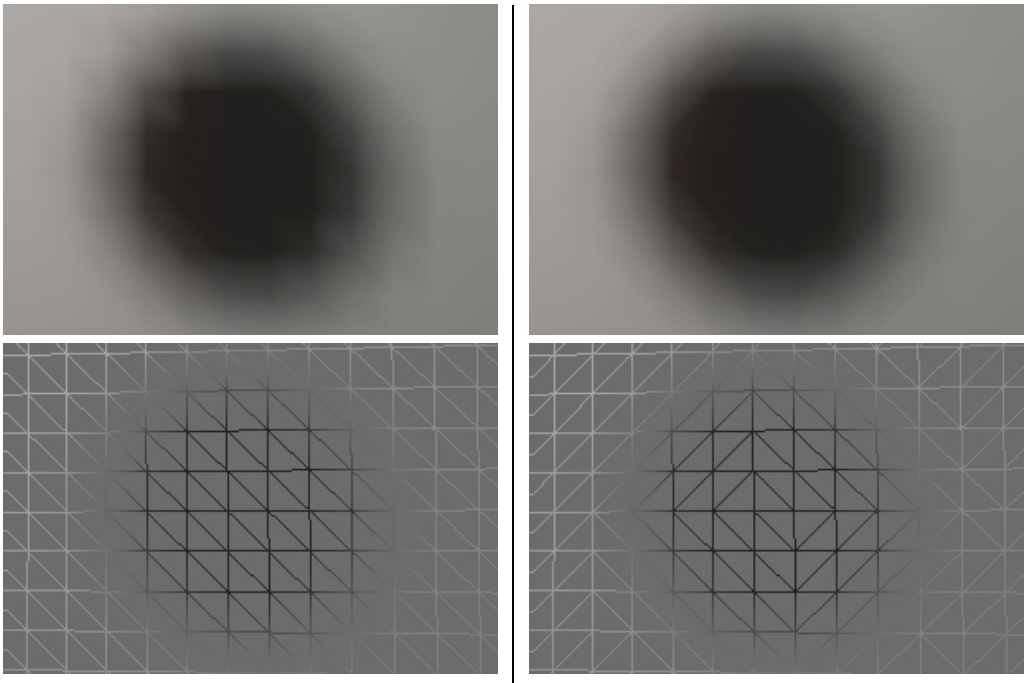
4.3 Suivi des ombres

Une fois le lissage effectué, il est possible de rendre la scène directement en utilisant les librairies d'affichages du type OpenGL. Mais cette librairie fait une triangulation uniforme des polygones (cf. Fig. 3.9-gauche), ne permettant pas une bonne représentation des variations d'intensité.

Nous proposons dans ce cas de simplement trianguler suivant la diagonale de plus petite variation (cf. Fig. 3.9-droite), afin de s'adapter à l'interpolation effectuée par le matériel graphique.

5 Conclusion

Tout ce que nous avons présenté ici, nous a permis de développer une plateforme d'expérimentation des algorithmes d'éclairage global. Par défaut, celle-ci contient un



triangulation OpenGL | gradient d'intensité

FIG. 3.9 : Suivie des ombres

algorithme de radiosité hiérarchique avec regroupement. L'architecture et les solutions déjà proposées ici, vont permettre de tester d'autres approches pour le contrôle de la qualité. Nous allons notamment développer une approche combinant cette solution de radiosité hiérarchique avec regroupement, avec d'autres méthodes de calcul des échanges globaux, afin d'offrir à l'utilisateur une solution complète pour la simulation d'éclairage.

4

Nouvelle méthode unifiée

Rien ne sert de courir, il faut partir à point.

Jean de la Fontaine

Pouvoir obtenir rapidement une solution d'éclairage global, en prenant en compte tous les types de réflexion, est un grand défi. Cela permettrait un retour rapide à l'utilisateur, facilitant ainsi le contrôle de la qualité d'une solution finale. Si les approches par éléments finis ou de radiosité peuvent permettre une simulation rapide pour certains types de scènes, les méthodes probabilistes sont actuellement trop lentes ou trop bruitées.

Nous présentons ici un nouvel algorithme, qui, en intégrant un lancer de particules, pour les réflexions à caractère directionnel, dans l'étape de transfert d'énergie d'un algorithme de radiosité hiérarchique lui prend en compte les réflexions diffuses, permet de traiter efficacement tous les types de propagation lumineuse. Cette approche permet un calcul rapide et une simulation de bonne qualité visuelle. Les transferts diffus sont ainsi non bruités, grâce à la radiosité hiérarchique, mais il est aussi possible d'obtenir rapidement des effets spéculaires comme des caustiques. Cet algorithme peut aussi efficacement traiter le problème de l'éclairage indirect et permet une transition progressive d'une solution rapide mais de faible qualité vers une solution de grande qualité mais plus lente, permettant ainsi un contrôle de la qualité.

1 Introduction

Pour pouvoir obtenir un éclairage d'un environnement virtuel qui semble réaliste, nous avons vu qu'il est nécessaire de simuler l'éclairage de façon globale. En effet,

les effets de l'éclairage indirect peuvent représenter une part non négligeable dans certaines scènes.

Pour les scènes purement diffuses, les méthodes récentes (cf. Chap. 2-Sec. 2.4) offrent la possibilité d'interagir avec la solution : calcul rapide et possibilité de mise à jour interactive pour le déplacement des objets *[DS97, Sha97]* par exemple.

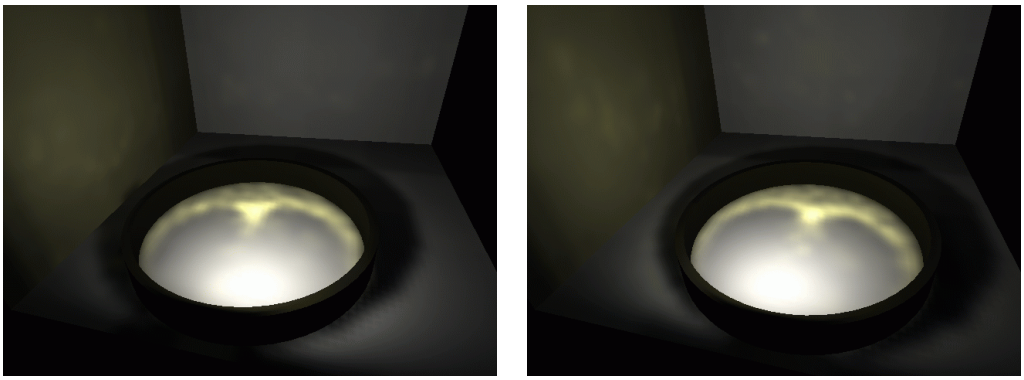
Cependant, les environnements réels sont rarement purement diffus, et en utilisant ces algorithmes, de nombreux effets lumineux sont ignorés. Les échanges lumineux non diffus étant en général fortement directionnels, une discrétisation de la fonction de radiance en fonction des directions se révèle rapidement trop coûteuse pour des scènes complexes. De plus, les algorithmes stochastiques, qui les prennent en compte, ont une convergence trop lente et souffrent trop du bruit pour une application interactive avec une bonne qualité visuelle. Il en va de même pour les méthodes en plusieurs étapes ("multi-pass"), qui combinent des solutions par éléments finis et stochastiques. Elles sont limitées soit par les effets lumineux possibles, soit par la complexité des scènes, ou requièrent des traitements complexes pour prendre en compte tous les chemins lumineux possibles.

Notre but est de fournir un algorithme automatique intégrant en tous les types de réflexion, permettant une simulation rapide et de grande qualité pour l'éclairage global. Pour y arriver, nous intégrons un lancer de particules (pour les échanges non diffus) dans une méthode de radiosité hiérarchique avec groupes ("clusters") qui nous permet un traitement efficace les échanges purement diffus. Cette intégration est simple et utilise l'information sur les chemins lumineux que représente les liens pour guider et accélérer l'émission de particules. La méthode qui en résulte est rapide et simule tous les chemins lumineux possibles $L(D|S)^*E$.

Pour des scènes dynamiques simples, notre algorithme permet une visualisation interactive de l'éclairage global (aux alentours de deux images par seconde sur une SGI Origin 200 avec un processeur à 200 Mhz), que cela soit de la composante diffuse ou directionnelle (cf. Fig. 4.1). Il peut aussi traiter des scènes plus complexes, avec un temps de calcul plus long. Le problème de l'éclairage indirect est aussi pris en compte par l'algorithme tel quel. Enfin, il possède par nature un mécanisme de contrôle de qualité, qui permet une transition progressive d'une solution rapide mais de basse qualité, vers une solution de grande qualité, mais qui requièrent plus de temps de simulation, ceci en changeant le niveau de la hiérarchie auxquels se font les échanges lumineux.

2 Aperçu général de la méthode

Notre nouvel algorithme peut être considéré dans son esprit comme une méthode en deux passes : une première passe calcule tout la solution indépendante du point de vue (chemins $L(D|S)^*D$), et un rendu final rajoute tous les effets qui eux dépendent du point de vue (chemins $(L|D)S^+E$).



Seul l'anneau se déplace (deux images par seconde sur une SGI R10000).
Notez la présence de la caustique en forme de cardioïde.

FIG. 4.1 : Deux images extraites d'une session interactive

2.1 Combiner les approches et répartir les calculs

Comme cela a été vu dans l'état de l'art, de nombreuses méthodes d'éclairage global ont été développées ces vingt dernières années, chacune étant efficace pour un ensemble de problèmes donnés. Par exemple, les algorithmes de radiosité hiérarchique avec regroupement sont très efficaces dans le calcul de solutions diffuses pour un environnement diffus. Mais elles se prêtent mal aux calculs d'échanges lumineux directionnels.

De même, les méthodes de lancer de rayons sont efficaces pour capturer les effets lumineux dépendants du point vue. Mais elles ont de la difficulté pour le calcul des effets lumineux dépendant des sources de lumières, comme les caustiques. De plus, la simulation des échanges diffus reste très bruitée ou trop lente, même dans le cas des méthodes bidirectionnelles.

En partant des sources de lumière, les méthodes de lancer de particules arrivent par contre à capturer assez bien les effets dépendant de ces sources. De plus l'étape de reconstruction permet d'atténuer l'influence du bruit. Mais cette étape reste lourde en temps de calcul.

C'est pourquoi, en utilisant la propriété additive de la lumière, il serait intéressant de combiner ces méthodes en un seul algorithme, afin de faciliter le calcul d'une solution complète : une radiosité hiérarchique avec regroupement pour les échanges diffus, et un lancer de particules pour capturer les caustiques et autres effets directionnels, et une lancer de rayons pour permettre la visualisation des effets dépendants du point de vue.

2.2 Utilisation de la structure hiérarchique

Les méthodes hiérarchiques ont montré leur efficacité dans le traitement des problèmes complexes, quel que soit le domaine. Pour obtenir une solution efficace face

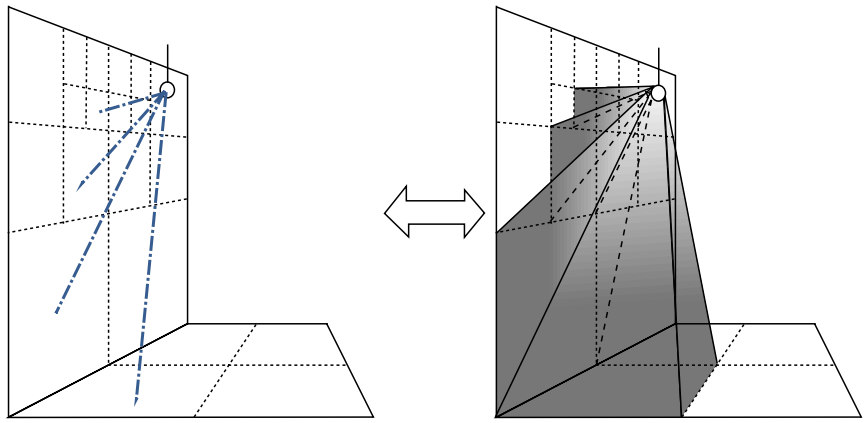


FIG. 4.2 : Espace des directions du aux liens

aux environnements réalistes, souvent très complexes, un traitement hiérarchique semble donc être une approche souhaitable. C'est ce qui oriente notre choix vers l'utilisation de la radiosité hiérarchique avec regroupement comme base principale de notre solution.

Mais surtout, ces algorithmes créent une structure, les liens, qui est assimilable à un partitionnement de l'espace des échanges possibles (cf. Fig. 4.2). Ainsi, il est possible de sélectionner les échanges pour lesquels un traitement directionnel est nécessaire, et de ce fait, de restreindre spatialement les lieux d'émission de particules, comme dans le "Photon Map" *[Jen96]*, et ce, de manière automatique ! De plus, sur ces liens sont sauvegardés des informations comme la visibilité entre deux éléments et le facteur de forme. Celles-ci peuvent être utilisées pour accélérer et guider l'émission de particules.

Ainsi, la principale contribution de notre approche est cette forte intégration des deux algorithmes, la radiosité hiérarchique avec groupes et un lancer de particules. Elle est simple et prend en compte par essence tous les chemins lumineux. Par rapport aux méthodes précédentes, l'utilisation d'un algorithme de radiosité hiérarchique permet d'obtenir une simulation de l'éclairage diffus efficacement et sans bruit, tout en permettant de détecter tous les types de transferts comme les caustiques indirectes. Cette approche offre un calcul rapide de tous les chemins lumineux, et une modification interactive pour les scènes dynamiques simples avec la prise en compte de tous les effets lumineux. Nous pouvons également simuler l'éclairage global pour des scènes complexes dans un temps plus long, mais raisonnable, et les cas difficiles comme un éclairage principalement indirect sont pris en compte sans traitement spécifique.

2.3 Une itération

Avant tout calcul sur la scène, la hiérarchie déjà existante est augmentée par une étape de regroupement, comme pour un algorithme de radiosité hiérarchique avec

groupe classique *[Sil95]*. La méthode de regroupement employée a été décrite dans le chapitre Chap. 3.

Raffinement

Le raffinement se fait en examinant chacune des interactions, représentée par un lien lorsqu'il a été établi au cours d'une itération précédente, en décidant s'il doit être subdivisé suivant le critère choisi. Cette étape est identique à celle des algorithmes de radiosité hiérarchique avec regroupement (cf. Chap. 2-Sec. 2.4). Notre critère de raffinement est basé sur l'énergie transportée au travers d'un lien, *BFA*, où *B* est la radiosité de l'émetteur, *F* le facteur de Forme (avec visibilité) et *A* le facteur d'aire *[Sil95]* du récepteur. Si cette énergie est supérieure à une quantité donnée, le lien doit être subdivisé (cf. Chap. 3-Sec. 3).

$$B_j F_{i,j} A_i \leq \varepsilon \tag{4.1}$$

Transfert d'énergie

Pendant cette étape, tous les liens sont parcourus pour transmettre l'énergie. Dans tous les cas, l'irradiance diffuse (I_d), résultant de la contribution de chaque lien, est calculée sur le récepteur, comme dans l'algorithme classique. Pour chacun des liens arrivant sur une surface ayant une composante directionnelle dans sa propriété de réflexion, des particules sont émises (cf. Sec. 3.1) pour simuler l'échange diffus vers non diffus. Un groupe est considéré comme diffus si tous les éléments qu'il contient sont diffus. Cette utilisation des liens permet de réduire l'espace vers lequel il faut envoyer des particules. Celles-ci sont alors propagées dans l'environnement par des réflexions directionnelles.

Ces particules sont immédiatement réfléchies sur le récepteur et retransmises dans l'environnement. Pour un groupe, la réflexion se fait sur le premier objet ou élément rencontré par celles-ci. Les particules sont alors propagées dans l'environnement jusqu'à ce quelles soient absorbées, réfléchies diffusément (car ce type de réflexion est pris en compte par la radiosité) ou qu'elles sortent de l'environnement. Chacun des impacts est stocké sur les objets, en excluant le premier récepteur, cet échange étant déjà pris en compte par les calculs d'irradiance sur les liens.

"Push/Pull" et placement des particules

Après la propagation, les particules sont toutes stockées au plus haut niveau des objets intersectés (comme des sphères, des polygones, cf. Chap. 3-Sec. 2.3 et Annexe A). Comme nous voulons utiliser les particules pour reconstruire l'irradiance non diffuse, et ceci de façon hiérarchique, il faut pouvoir décider à quel niveau il est judicieux les placer. Cette étape est une mise à jour de la représentation hiérarchique : les particules sont placées au juste niveau hiérarchique et leur énergie est ajoutée à l'irradiance

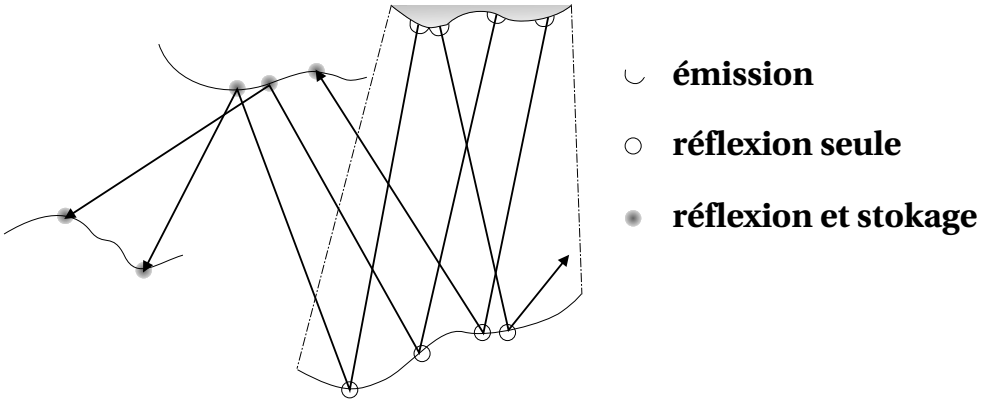


FIG. 4.3 : Émission et propagation des particules

diffuse. Cette intégration permet de tenir compte simplement de la contribution des particules, et de traiter ainsi efficacement tous les chemins $L(D|S)^*D$.

Pour l'affichage, une étape de lissage de la solution de radiosité est nécessaire. Ensuite, la solution peut être rendue interactivement en utilisant OpenGL (solution $L(D|S)^*DE$), ou par un lancer de rayons (solution $L(D|S)^*E$), ou une combinaison des deux (un lancer de rayons ne calculant que $(L|D)S^*E$).

3 Intégration des deux algorithmes

Pour pouvoir lancer des particules afin de simuler des transferts non diffus, nous allons introduire une étape modifiée de transfert, dans laquelle les particules sont émises et propagées dans l'environnement, suivie d'une étape modifiée de "Push/Pull", dans laquelle les particules sont positionnées au juste niveau hiérarchique. Comme les particules sont émises dans un espace restreint défini par un lien, il faut aussi pouvoir déterminer le nombre de particules à envoyer, ainsi que leur énergie propre.

L'initialisation de l'algorithme est identique à celle d'une méthode hiérarchique avec regroupement : la hiérarchie avec groupes est tout d'abord construite en utilisant l'approche présentée dans le chapitre Chap. 3, puis un lien est créé du sommet de la hiérarchie vers lui-même *[Sil95]*. Pour pouvoir tenir compte des transferts non diffus, une fonction de réflexion généralisée, ou BSDF *[Vea97]*, est stockée sur chaque élément hiérarchique. Nous utilisons la représentation mixte diffus/directionnel du modèle *[LW94]*, qui permet de choisir stochastiquement si une réflexion sera diffuse ou pas. Ce choix se fait sur le pourcentage diffus/directionnel défini par la propriété du matériau. Seul un groupe ("cluster") ne contenant que des éléments diffus peut être considéré comme diffus.

3.1 Émission de particules

Une grande partie de l'algorithme décrit ci-dessus réside dans l'émission et la réflexion des particules. Nous allons maintenant décrire comme prendre en compte l'énergie qui passe à travers un lien à l'aide des particules. Ces particules vont devoir représenter une bonne estimation du flux circulant à travers ce liens.

Pour estimer stochastiquement le flux émis par une surface (cf. Chap. 2-Eq. 2.4), c'est à dire

$$\Phi_S^{s+} = \int_S \int_{\Omega^+(s)} L(s,\omega) < \omega \cdot n > d\omega ds$$

ceci par n_p particules émises selon une densité probabilité $p(s,\omega)$, chacune d'elles doit avoir une énergie de *[Vea97, Wal98]* :

$$\phi_i = \frac{L(s_i,\omega_i) < \omega_i \cdot n_{s_i} >}{n_p \; p(s_i,\omega_i)} \tag{4.2}$$

Ainsi, l'estimateur du flux émis $\Phi_{n_p} = \sum_{i=1}^{n_p} \phi_i$ est sans biais (cf. Chap. 2) :

$$\begin{aligned}
E[\Phi_{n_p}] &= E\left[\sum_{i=1}^{n_p} \frac{L(s,\omega)<\omega \cdot n_s>}{n_p \; p(s,\omega)}\right] \\
&= \frac{1}{n_p} \sum_{i=1}^{n_p} E\left[\frac{L(s,\omega)<\omega \cdot n_s>}{p(s,\omega)}\right] \\
&= \frac{1}{n_p} \sum_{i=1}^{n_p} \int_S \int_{\Omega^+(s)} \frac{L(s,\omega)<\omega \cdot n_s>}{p(s,\omega)} p(s,\omega) d\mu(x,w) \\
&= \int_S \int_{\Omega^+(s)} L(s,\omega) < \omega \cdot n_s > d\omega ds \\
&= \Phi_S^{s+}
\end{aligned}$$

Considérons maintenant l'énergie $L(s,r)$ quittant le point s d'un émetteur S et passant au point r d'un récepteur R. En utilisant la notation de Sillion *[Sil95]*, dans le cadre d'une hiérarchie générale, le flux entre deux éléments hiérarchiques est :

$$\Phi_{RS} = F_R F_S \int_S \int_R L(s,r) \frac{\tau R_r E_s}{||s-r||^2} dr \, ds \tag{4.3}$$

où R_r est 1 pour les groupes et $\cos\theta_i$ pour les surfaces, E_s est 1 pour les groupes et $\cos\theta_o$ pour les surfaces, F_S, F_R sont des facteurs d'échelles pour tenir compte des cas volumiques, valant 1 pour les surfaces et $\frac{1}{4}$ pour les groupes . Enfin, τ est l'absorption pour des milieux participants. Les valeurs $\cos\theta_i$ et $\cos\theta_o$ sont les cosinus de l'angle entre la direction de propagation et la normale respectivement de l'émetteur et du récepteur (cf. Chap. 2-Sec. 2.4 et Chap. 2-Tab. 2.2).

Dans le cadre d'un algorithme de radiosité, $L(s,r)$ s'exprime sous la forme :

$$L(s,r) = \frac{B_S}{\pi} \tag{4.4}$$

Ainsi, l'équation (4.3) devient

$$\Phi_{RS} = B_S F_{RS} A_R \tag{4.5}$$

où B_S est la radiosité de S et F_{RS}, le facteur de forme (sans la visibilité) entre S et R. A_R est le facteur d'aire défini dans *[Sil95]* (cf. Tab. 2.2). Si le flux Φ_{RS} est estimé avec n_{RS} particules d'énergie constante ϕ_{ct}, sa valeur est alors :

$$\Phi_{RS} = n_{RS}\phi_{ct} \tag{4.6}$$

Nous pouvons alors choisir le nombre de particules à envoyer à l'aide de l'égalité suivante :

$$n_{RS} = \left\lfloor \frac{B_S A_R F_{RS}}{\phi_{ct}} \right\rfloor \tag{4.7}$$

Pour un lien donné, le facteur de forme, le facteur d'aire, et la radiosité de la source sont des données connues. Mais ce nombre ne permet pas de prendre en compte la totalité du flux à cause de l'approximation nécessaire vers une valeur entière. Le résidu sur le flux est alors

$$res = \frac{B_S A_R F_{RS}}{\phi_{ct}} - n_{RS} \tag{4.8}$$

Nous pouvons constater ainsi que *res* appartient à l'intervalle $[0,1[$. Ce résidu est ainsi considéré comme la probabilité d'émission d'une particule supplémentaire d'énergie ϕ_{ct}.

Dans le cas de particules d'énergie constante, la loi de probabilité d'émission est alors, en utilisant les équations Eq. (4.2) et (4.3), et dans le cas d'un radiosité constante par élément :

$$\begin{aligned} p(s,r) &= \frac{L(s,r)\ F_R\ F_S\ \tau\ R_r\ E_s}{\|s-r\|^2} \frac{1}{n_{RS}\ \phi_{ct}} \\ &= \frac{\frac{B_S}{\pi}\ F_R\ F_S\ \tau\ R_r\ E_s}{\Phi_{RS}\ \|s-r\|^2} \\ &= \frac{B_S\ F_R\ F_S\ \tau\ R_r\ E_s}{\pi\ B_S\ F_{RS}\ A_R\ \|s-r\|^2} \\ p(s,r) &= \frac{F_R\ F_S\ \tau\ R_r\ E_s}{\|s-r\|^2\ \pi\ F_{RS}\ A_R} \end{aligned} \tag{4.9}$$

Deux lois dérivent de cette dernière, une pour choisir un point s sur l'émetteur S ($p'(s)$) et l'autre pour choisir un point r sur le récepteur R, connaissant déjà le point s_0 sur S, soit $p'(r/s = s_0)$:

$$\begin{cases} p'(s) &= \frac{F_{sR}}{A_R F_{RS}} \\ p'(r/s = s_0) &= \frac{<(r-s_0)\cdot n_{s_0}><(s_0-r)\cdot n_r>}{\pi F_{s_0 R}\|r-s_0\|^4} \end{cases} \tag{4.10}$$

Dans ces équations, F_{sR} est le facteur de forme du point s vers l'élément R. Ces équations sont réciproques et il est possible de choisir d'abord un point r sur le récepteur puis sur la source s. Les équations précédentes deviennent :

$$\begin{cases} p'(r) &= \frac{F_{rS}}{A_S F_{SR}} = \frac{F_{rS}}{A_R F_{RS}} \\ p'(s/r=r_0) &= \frac{<(r_0-s)\cdot n_s><(s-r_0)\cdot n_{r_0}>}{\pi F_{r_0 S}\|r_0-s\|^4} \end{cases} \tag{4.11}$$

Malheureusement, l'utilisation de ces équations se révèle très complexe, voire impossible. En effet, elles nécessitent la possibilité de pouvoir calculer de manière exacte le facteur de forme sur des objets quelconques. Dans ce contexte, il est préférable d'utiliser des particules d'énergies différentes. Mais cela crée des solutions beaucoup plus bruitées (la précision des méthodes de reconstruction ne dépend en général que de la densité de particules).

Ainsi, pour chacune des n_{RS} particules, un point s et un point r sont déterminés par une loi uniforme sur respectivement S et R, $p(r) = \frac{1}{\mu(R)}$ et $p(s) = \frac{1}{\mu(S)}$ où $\mu()$ est l'aire pour une surface et le volume pour un groupe. Les deux tirages étant indépendants, la probabilité de choisir un couple points (r,s) est alors :

$$p(r,s) = \frac{1}{\mu(R)\ \mu(S)} \tag{4.12}$$

L'énergie de la particule devient ainsi (cf. Eq. 4.2) :

$$\phi_{rs} = \frac{L(s,r) S_R S_S \mu(R)\ \mu(S) \tau R_r E_s}{\|s-r\|^2 n_{RS}} \tag{4.13}$$

Toutes les particules peuvent être absorbées ou réfléchies de manière diffuse, si le récepteur possède une composante diffuse et directionnelle dans sa BSDF. Ces transferts étant pris en compte par la radiosité, ces propagations ne sont pas effectuées. Nous ne propageons et stockons que les particules qui sont réfléchies selon la composante directionnelle.

3.2 Placement hiérarchique des particules

Les effets spéculaires correspondent souvent à des hautes fréquences dans la solution d'éclairage. Intuitivement, cela correspond à des zones de forte concentration de particules. Pour les détecter, nous utilisons un "critère d'étalement" : si la concentration énergétique des particules représente une faible proportion du facteur d'aire de l'élément les supportant (cf. Fig. 4.4), alors il faut descendre les particules dans la hiérarchie.

Lors de la traversé de la hiérarchie, pour un élément nous avons n particules, chacune d'énergie ϕ_i et à la position x_i. Nous définissons un centre c comme la position moyenne pondérée par l'énergie des particules. Nous définissons aussi un "critère

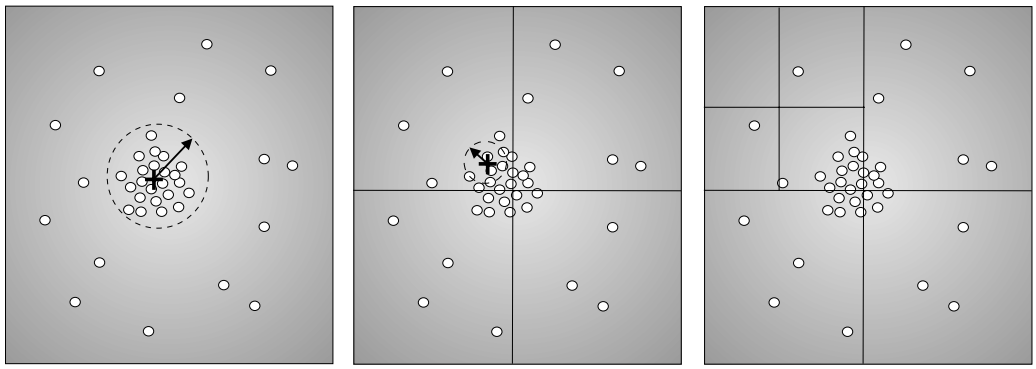


FIG. 4.4 : Placement hiérarchique des particules

d'étalement" CE comme suit :

$$c = \frac{\sum_n \phi_i x_i}{\sum_n \phi_i}, \quad CE = \frac{\sum_n \phi_i \|c - x_i\|^2}{\sum_n \phi_i} \tag{4.14}$$

Si A est le facteur d'aire *[Sil95]* de l'élément courant, le test est vérifié si

$$\pi CE \leq \zeta A \; et \; \sum_{i=1..n} \phi_i \geq \varepsilon \tag{4.15}$$

où ζ marque l'objectif de concentration à atteindre. Par expérience, nous utilisons la valeur $\zeta = 0.5$. Intuitivement, cette condition est atteinte lorsqu'il y a une forte concentration de particules de haute énergie comparée à la taille de l'élément traité. Dans le cas contraire, les particules restent à ce niveau, sinon, elles sont descendues dans la hiérarchie. Une subdivision plus fine peut alors se faire, adaptant automatiquement le maillage aux détails fins des effets spéculaires. Les liens restant à leur niveau, il n'en résulte pas un surcoût de calcul pour la radiosité.

3.3 Intégration des particules à la solution de radiosité

Lorsque les particules sont positionnées au juste niveau hiérarchique, il convient d'ajouter leur contribution à l'irradiance. Chaque élément de la hiérarchie possède déjà une irradiance I_d due aux échanges diffus à travers les liens, calculée pendant l'étape de transfert, et n particules ayant chacune une énergie ϕ_i. Son facteur d'aire étant A, l'irradiance totale I devient :

$$I = I_d + \frac{\sum_n \phi_i}{A} \tag{4.16}$$

Cela peut être vu grossièrement comme une reconstruction hiérarchique de l'irradiance due aux particules. Sur les feuilles de la hiérarchie, la valeur de radiosité devient ainsi $B = R_d I$, comme pour une radiosité diffuse pure. La radiosité diffuse obtenue reste constante sur les feuilles de la hiérarchie. La représentation hiérarchique est alors mise à jour en remontant les valeurs de radiosité et en les moyennant.

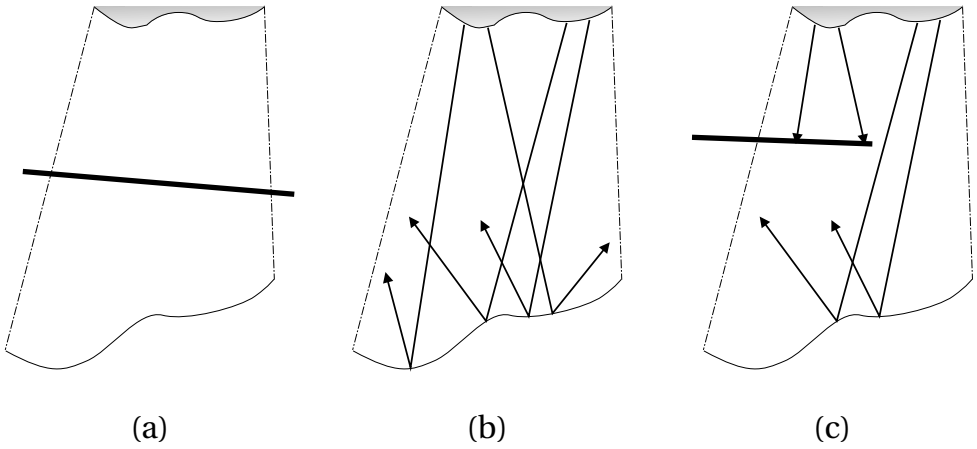


FIG. 4.5 : Utilisation de l'information de visibilité

Notez qu'il est possible d'accumuler l'irradiance due aux particules en faisant une moyenne sur les itérations. L'expérience a montré que cette approche améliorait la qualité finale en réduisant le bruit. Cela revient à combiner les estimateurs de chaque itération *[Vea97, SW99]*.

4 Accélération de la simulation

Le processus de raffinement nous fournit les informations de visibilité qui seront utilisées pour accélérer un lancer de particules (cf Fig. 4.5). Lorsque qu'une particule est envoyée par un lien dont la visibilité est totale, aucun test n'est fait pour déterminer s'il atteint bien le récepteur. De même, aucune particule n'est envoyée si la visibilité est nulle. Cette restriction n'est possible que si une visibilité nulle (ou totale) exacte est disponible. Dans le cas contraire, cela induirait une dégradation de la qualité finale.

De plus, l'utilisation d'une représentation par objets (cf Chap.3-Sec.2.3) nous permet de réduire le coût de calcul et d'accroître la qualité. En effet, ces objets ne sont pas présubdivisés, accélérant ainsi le calcul de lancer de rayons et améliorant aussi la qualité des ombres et des caustiques.

Enfin, pour les scènes dynamiques, le recalcul nécessaire est limité. Lorsqu'un objet quelconque se déplace, aucune subdivision n'est détruite. Une fois que le mouvement est effectué, la hiérarchie est traversée pour recalculer la visibilité pour tous les liens. Seuls les facteurs de forme (sans la visibilité) des liens atteignant ou partant de l'objet en mouvement sont recalculés. Les particules sont alors émises de nouveau. Le principe de mise à jour nécessite que l'irradiance provenant des réflexions spéculaires ne soit pas accumulée. Pour une meilleure interactivité, le nombre de rayons lancés pour le calcul de la visibilité est réduit. Un traitement plus complet du cas dynamique sera fait dans le chapitre suivant (cf. Chap. 5).

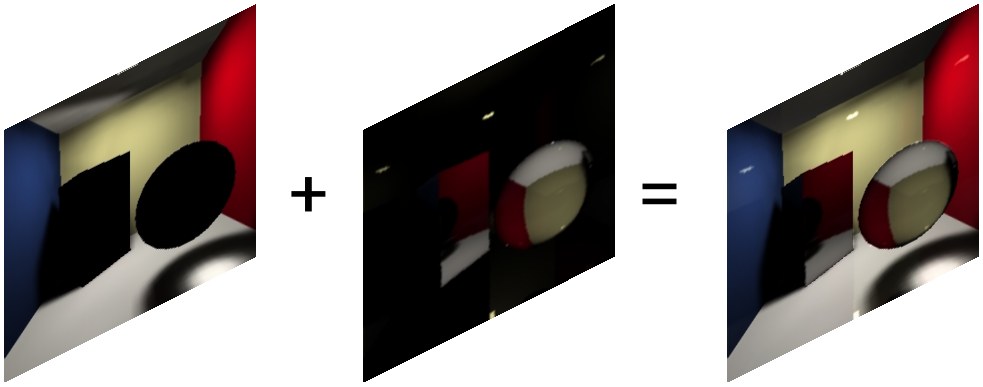


FIG. 4.6 : Utilisation du Render Cache *[WDP99]*

5 Reconstruction et rendu

Notre algorithme calcule une valeur de radiosité constante par élément. Pour afficher une solution continue, une étape de reconstruction est nécessaire. Cette solution lissée peut être alors rendue directement grâce au matériel graphique. Pour avoir les chemins $(L|D)S^+E$ manquants, un lancer de rayons est utilisé.

5.1 Lissage

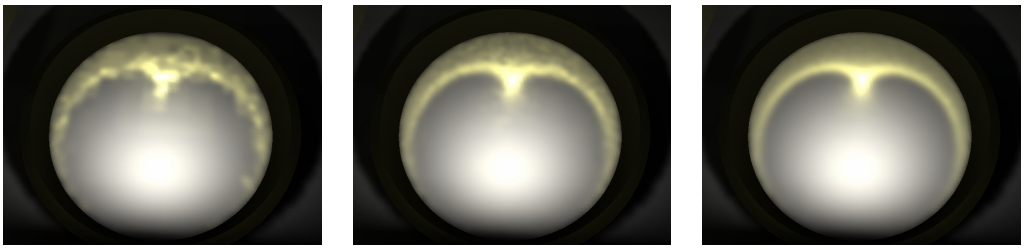
Comme première approche, nous reconstruisons une “grille” séparément des objets de la hiérarchie (cf. Chap. 3-Sec. 4.1). Elle représente un ensemble de polygones à afficher, correspondant à un maillage complet pour la subdivision maximale atteinte. Bien sûr, des reconstructions plus performantes *[KSS97]* peuvent être adaptées à nos besoins (cf. Sec. 7).

Une fois calculée, cette grille peut être rendue directement par le matériel graphique. Les seuls chemins manquant sont les chemins $(L|D)S^+E$.

5.2 Ajout de la composante dépendante du point de vue

Pour tenir compte de ces chemins, nous utilisons un lancer de rayons pour ajouter la composante $(L|D)S^+E$ manquante au rendu effectué par le matériel graphique qui correspond à la composante diffuse seule de l’image. A chaque intersection (sauf la première), l’objet intersecté est interrogé pour récupérer la valeur diffuse au point considéré. Cette valeur est une interpolation des valeurs aux sommets du polygone de grille concerné.

Ce lancer de rayons peut être fait interactivement en utilisant le “Render Cache” *[WDP99]*. La scène est alors rendue dans un premier temps en ignorant la transparence qui sera prise en compte par un lancer de rayon. La composante dépendante du point



4 secs/1 200 particules 5 secs/7 800 particules 15 secs/81 800 particules

11 primitives. Le nombre de particules et le temps nécessaire à deux itérations sont indiqués. L'aire minimale reste constante. L'image est rendue en OpenGL.

FIG. 4.7 : Variation du paramètre ϕ_{ct}

de vue calculée par le "Render Cache" *[WDP99]* est considérée comme une texture appliquée à un polygone au premier plan, occupant toute la fenêtre. La sommation des deux composantes se fait alors sur la carte graphique (`glBlendFunc(GL_ONE, GL_ONE)`, cf. Fig. 4.6).

6 Résultats

Tous les résultats présentés ici ont été obtenus sur une Origin 2000 (SGI) munie de processeurs R10000 à 200 Mhz.

6.1 Rendu interactif pour les scènes simples

Un exemple est montré dans la figure 4.1. Dans cette scène très simple (un anneau spéculaire, des murs diffus et une source diffuse), il est possible d'interagir avec une mise à jour de l'éclairage global. L'absence de reflets est due à un rendu OpenGL seulement. La figure 4.11-(a) en montre un autre exemple une peu plus complexe. En effet il est composé d'une tasse (plus complexe qu'un anneau), de murs et d'une lampe diffuse. Il est ici encore possible d'avoir une mise à jour interactive. Un film au format Quicktime d'une session interactive peut être trouvé à :
`http ://www-imagis.imag.fr/Publications/xgranier/Thesis`.

Ces exemples montrent ainsi l'efficacité d'une telle approche. En effet, il n'y a pas vraiment ici d'algorithme conçu pour le cas dynamique, c'est une simple adaptation ne réduisant que peu les calculs à effectuer.

6.2 Contrôle de qualité

La simulation est dirigée par trois paramètres :

– (i) un seuil ε pour le critère de raffinement *BFA* qui contrôle le niveau où se trouvent les liens dans la hiérarchie, et donc la qualité de l'éclairage diffus.

La simulation a pris 17 min 22 s.

FIG. 4.8 : Deux vues d'une même scène.

– (ii) l'aire minimale : elle détermine la finesse du résultat par la taille minimale d'un élément de la hiérarchie.
– (iii) le paramètre ϕ_{ct} contrôle le nombre de particules à envoyer, et donc la qualité de l'éclairage spéculaire (cf. Fig. 4.7). Plus ϕ_{ct} est petit, et plus le nombre de particules à envoyer à travers un lien est grand (cf. Eq. 4.7), et plus les effets directionnels seront bien capturés.

6.3 Scènes complexes et éclairage indirect

Pour la figure 4.8, nous avons modifié la scène "Soda Shoppe"[1] en réduisant la taille des sources et en ajoutant des sphères en verre bleu pour mieux visualiser la présence de caustiques. Cette scène contient 1 779 primitives (soit environ 80 000 polygones pour une subdivision de niveau 5). Notez que, dans un temps raisonnable (17 minutes 22 sec), des caustiques ont été obtenues. Un lancer de rayons a été fait pour obtenir les figures 4.8 et 4.9 en 4 minutes environ (image $1980x980$).

Dans la figure 4.9 (à gauche), nous avons la même scène, mais avec un éclairage indirect. Les sources de lumières sont dirigées vers les murs droit et gauche. Une telle scène serait très difficile à traiter avec une méthode nécessitant le reclassement des sources secondaires *[CRMT91]*. Notre algorithme l'a simulé en 9 minutes 2 sec, ce qui reste dans l'ordre de grandeur du cas direct (Fig. 4.8).

6.4 Comparaison informelle avec un lancer de particules

Cette dernière scène permet aussi une comparaison informelle entre un lancer de particules standard *[Wal98]* et notre méthode. Pour un temps de calcul équivalent, et avec la même méthode de reconstruction, un lancer de particules reste très bruité. En effet, l'éclairage de cette scène est principalement indirect et diffus. C'est une des configurations que les méthodes probabilistes ont le plus mal à traiter.

[1] disponible sur `http ://floyd.lbl.gov/mgf/scenes.html`

Notre méthode
9 minutes 2 sec

Lancer de particules
8 minutes 2 sec

FIG. 4.9 : La scène Soda Shoppe scène avec un éclairage indirect

6.5 Visualisation interactive et “Render Cache” *[WDP99]*

Nous avons intégré notre méthode avec le “Render Cache” *[WDP99]* pour une visualisation interactive des composantes directionnelles de l’image. Notre algorithme s’adapte bien à cette approche, car une surface diffuse est rendue directement en utilisant OpenGL, permettant de conserver une visualisation de la solution diffuse sans difficultés tout au long du mouvement. Deux images ont été capturées au cours d’une session interactive pour la figure 4.10 et cette session est disponible dans le film sur la page Web.

7 Conclusion

Nous avons présenté un nouvel algorithme pour l’éclairage global, qui prend en compte les effets diffus comme directionnels, d’une manière simple et intégrée.

Cela a été possible par l’automatisation de l’intégration d’un lancer de particules pour les échanges directionnels, dans l’étape de transfert d’énergie d’un algorithme de radiosité hiérarchique. La radiosité permet de simuler efficacement les interactions diffuses. De plus, les liens crées contiennent de nombreuses informations sur le flux lumineux à travers la scène. Nous avons exploité ces informations pour guider et accélérer l’émission de particules qui simule les interactions directionnelles. Cela permet une simulation de tous les chemins lumineux, comprenant les réflexions spéculaires (créant des caustiques) sans le coût et le bruit d’un lancer de particules pour toutes les interactions.

Notre méthode intégrée peut traiter efficacement les cas d’éclairage indirect diffus comme directionnel. Il permet aussi une transition progressive entre une solution rapide mais de faible qualité, vers une solution plus lente mais de grande qualité, sans pour autant omettre des chemins lumineux.

$3,5$ Hz (SGI R12000 400Mhz).
Dans un déplacement vers la droite, des défauts apparaissent là où les échantillons ne sont plus valides.

FIG. 4.10 : Deux images d'une session interactive avec le "Render Cache" *[WDP99]*.

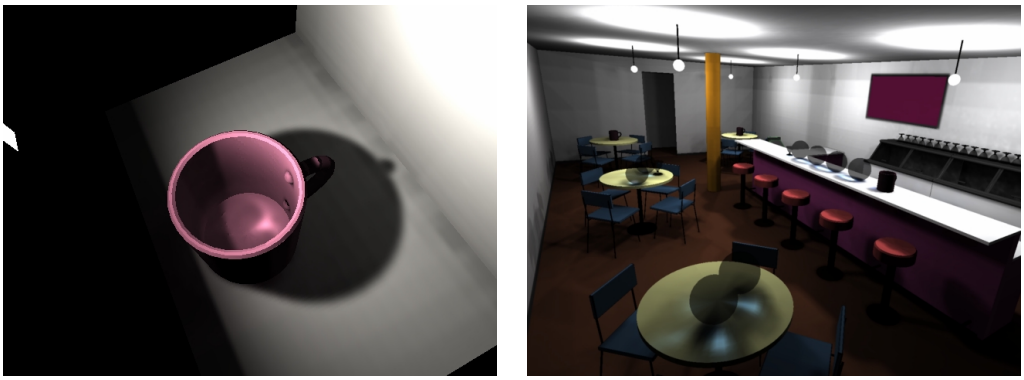
Une étape de reconstruction efficace permet de visualiser interactivement la lumière diffuse avec les réflexions spéculaires créant des caustiques. Cela peut être simulé interactivement pour les scènes dynamiques simples. Pour les scènes plus complexes, le temps de calcul reste raisonnable.

Pour un rendu de plus grande qualité, un lancer de rayons permet de visualiser correctement la composante directionnelle de certaines surfaces. Cela peut être fait interactivement en utilisant le "Render Cache" *[WDP99]*.

Limitations

Cependant, cette méthode unifiée ne constitue qu'une première approche de la mise en place d'un algorithme d'éclairage global automatique. Une de ses limitations est la nécessité de la création d'un maillage très fin pour visualiser correctement les hautes fréquences souvent présentes dans les phénomènes directionnels comme des caustiques. Le nombre d'éléments de maillage a tendance à exploser. Nous allons présenter dans le chapitre suivant une solution consistant en la reconstruction séparée des hautes fréquences (principalement spéculaires) et les basses fréquences (principalement diffuses). Les basses fréquences restent représentées dans le maillage, alors que les hautes fréquences sont traitées par l'utilisation de textures. Nous avons développé un algorithme de reconstruction hiérarchique pour les composantes basses fréquences, avec le traitement des hautes fréquences comme des textures. Nous le présenterons dans le chapitre suivant (cf. Chap. 5).

Pour la mise à jour interactive, nous présenterons une méthode étendant l'algorithme *[DS97]* aux cas des particules. En effet, un des points essentiels, est de parvenir



(a) scène avec une tasse (b) “Soda Shoppe”

Pour la scène (a), la source lumineuse peut être déplacée à $\sim$ 2 secondes par image. La transparence est simple, sans réfraction.

FIG. 4.11 : Promenade interactive par rendu OpenGL

à émettre des particules localement pour mettre à jour seulement les changements notables intervenus. Notre but est de pouvoir faire des mises à jour rapides, voire interactive, dans des scènes plus complexes.

5

Mise à jour incrémentale

Le luxe, c'est le temps

Jean-Jacques Goldman

Dans ce chapitre, nous allons présenter des techniques nouvelles pour essayer de résoudre certaines limitations de l'algorithme unifiée présenté précédemment. Nous introduirons ainsi une reconstruction différenciée pour les hautes fréquences et les basses fréquences de la solution d'éclairage global, calculé avec l'algorithme du chapitre précédent. Cette séparation se fait de manière automatique à l'aide de paramètres globaux.

Nous présenterons aussi une structure spatiale dynamique permettant de restreindre le nombre de particules renvoyées pendant les modifications locales d'une mise à jour incrémentale. L'association avec une structure de liens hiérarchique pour la solution diffuse, nous permettra de calculer rapidement des mises à jour sur des scènes complexes. L'objectif final est l'interactivité pour permettre la modélisation de scènes d'éclairage global, mais surtout, pour que l'utilisateur puisse avoir un contrôle sur le résultat final, c'est à dire qu'il puisse positionner les éléments d'une scène pour obtenir les effets souhaités.

1 Introduction

1.1 Motivations

L'algorithme que nous avons présenté dans le chapitre précédent possède un fort potentiel dans le traitement rapide des scènes complexes pour l'éclairage global. Cependant, comme nous l'avons signalé, il reste peu adapté à la mise à jour incrémentale,

voir interactive, nécessaire pour pouvoir faire de la modélisation d'environnements dans un contexte d'éclairage global. Les actions à envisager sont par exemple le placement d'objets pour contrôler les effets dans solution finale. Il est important de pouvoir effectuer ces tâches rapidement.

Bien que de nombreux travaux aient été effectués dans le cadre diffus *[Che90, GSG90, DS97, Sha97]*, permettant le déplacement interactifs d'objets, il n'existe pour l'heure aucune méthode permettant d'interagir avec une scène contenant des effets diffus comme directionnels, afin d'étudier l'influence du déplacement sur la solution complète.

En effet, le traitement des effets globaux directionnels, comme des caustiques, par exemple, reste difficile même pour des scènes modérément complexes. La plupart des approches existantes sont basées soit sur un lancer de rayons *[Jen96]*, et dépendent donc du point de vue et n'autorisent pas une promenade interactive ("walk-through"), soit sur un lancer de particules *[WHSG97]*, qui permet le calcul d'une solution indépendante du point de vue. Le traitement de tout les échanges par de telles méthodes rend difficile l'interaction avec la solution. Même les algorithmes récents, qui permettent une certaine interaction avec l'environnement pour un lancer de rayons *[PMS+99, WDP99]* restent limités dans la résolution de l'image et dans les effets lumineux qu'ils sont capables de traiter.

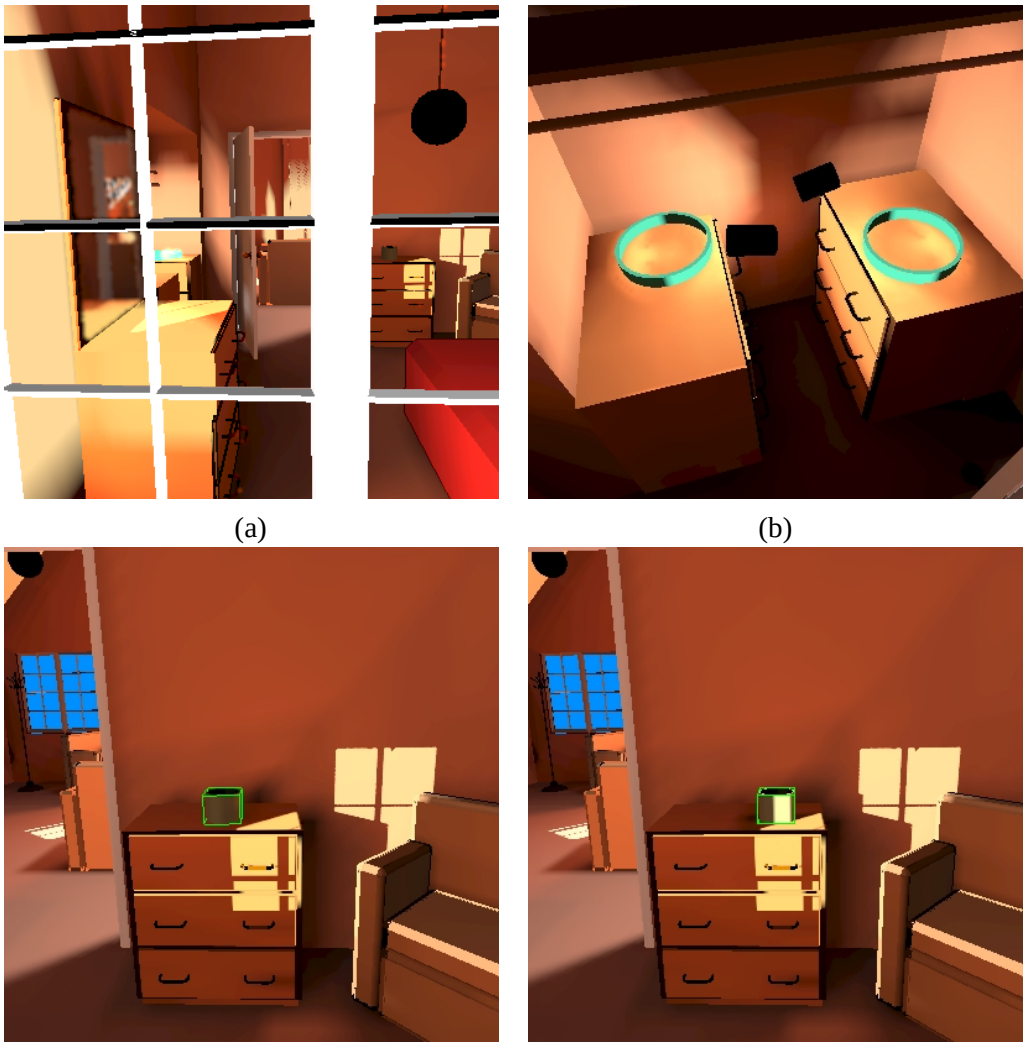
La solution que nous avons présentée dans le chapitre précédent peut nous aider à résoudre ce problème. En effet, en combinant les avantages d'une méthode de radiosité hiérarchique pour les échanges diffus et la restriction de l'émission des particules pour les échanges directionnels, cette approche peut permettre une mise à jour rapide.

Cependant la méthode unifiée conserve certaines limitations, comme la limitation à des scènes modérément complexes. En effet, l'utilisation des éléments pour la reconstruction des effets hautement directionnels conduit à une forte subdivision du maillage de la solution et donc une forte consommation mémoire. Nous allons présenter une représentation alternative à partir de textures pour résoudre ce problème.

La consommation de place mémoire est aussi une des limitations des algorithmes dynamiques diffus *[DS97]*. Pour pouvoir l'appliquer à des scènes complexes, il faut pourvoir réduire ce surcoût.

1.2 Positionnement de notre méthode

Les deux méthodes les plus liées à notre approche sont les "cartes de photon" ("Photon Map") de Wann Jensen *[Jen96]* et les travaux sur l'estimation de densité pour l'éclairage global de Walter et al. *[WHSG97]* (cf. Chap. 2). Les méthodes de reconstruction décrites dans le Photon Map sont dépendantes du point de vue mais assez simples d'implémentation et assez efficaces, malgré des temps de calculs qui restent conséquents. D'un autre coté, les techniques d'estimation de densité fournissent un cadre théorique et précis à la reconstruction. Bien que les solutions proposées soient indépendantes du point de vue, elles restent relativement lourdes en temps de calcul.



(a) (b)

(c) (d)

(a) Solution d'éclairage global. L'éclairage provient de lampes à l'intérieur de la maison ainsi que du soleil qui passe à travers la fenêtre et se réfléchit sur le miroir à gauche. Il éclaire ainsi indirectement le mur du fond et la chaise sur la droite. (b) Détail des caustiques dans le placard à gauche. (c)-(d) Déplacement du récipient sur la commode.

FIG. 5.1 : Quelques résultats sur une scène l'intérieur

Nous allons essayer de combiner les deux approches afin de développer un algorithme pouvant reconstruire les données directionnelles relativement rapidement.

De nombreuses techniques ont été développées pour permettre une visualisation, voir même une modification, d'une solution d'éclairage, que cela soit par l'utilisation des accélérations graphiques matérielles et des méthodes en plusieurs étapes (Diefenbach et al. *[DB97]*, et Stürzlinger et Bastos *[SB97]*), ou par des structures permettant un

lancer de rayons en temps interactif *[PMS+99, WDP99, WSBW01]*. Mais ces algorithmes souffrent d'une limitation sur les phénomènes lumineux possibles.

Comme nous le présenterons dans les sections qui suivent, nous utiliserons une structure de subdivision spatiale pour les mises à jour incrémentales. De telles structures ont été employées récemment pour accélérer le lancer de rayons interactif pour des scènes dynamiques *[RSH00]*. Nous utiliserons aussi des textures pour représenter les caustiques et les effets directionnels. Cette approche est similaire à celles présentées par Heckbert avec ses textures de radiosité *[Hec90]*, de même que celles présentées par Myszkowski et Kunii *[MK94]*, et par Bastos et al. *[BGB97]* entre autres. Mais notre cadre d'application est différent (limité à la reconstruction de certains effets directionnels et pour une combinaison avec un maillage) et nécessite ainsi de nouvelles approches.

Radiosité hiérarchique dynamique

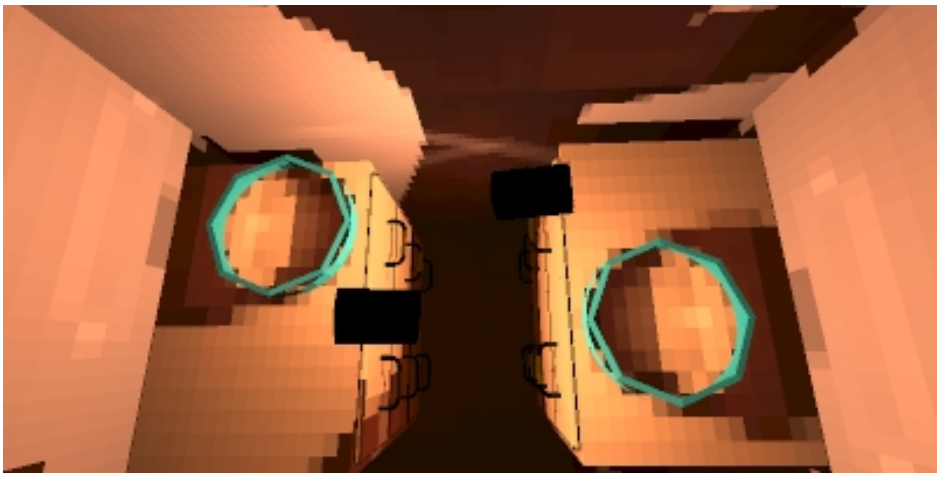
Les méthodes de radiosité hiérarchique ont montré leur capacité d'adaptation aux cas dynamiques. Nous utiliserons ces techniques basées sur une hiérarchie de liens *[DS97]* pour mettre à jour efficacement la composante diffuse de l'algorithme unifié.

En effet, les liens d'un algorithme de radiosité hiérarchique classique induisent une partition des échanges lumineux entre les éléments d'une scène. Drettakis et Sillion *[DS97]* ont augmenté l'information disponible sur ces liens par l'utilisation d'un volume d'interaction ("shaft" *[HW94]*). Cette structure permet de détecter efficacement les liens affectés par le mouvement d'un objet. Ainsi, une mise à jour locale est possible, et seule certaines zones de la hiérarchie nécessitent une modification. Il ne reste dans notre approche plus qu'à traiter le cas des échanges directionnels, notamment une émission localisée des particules.

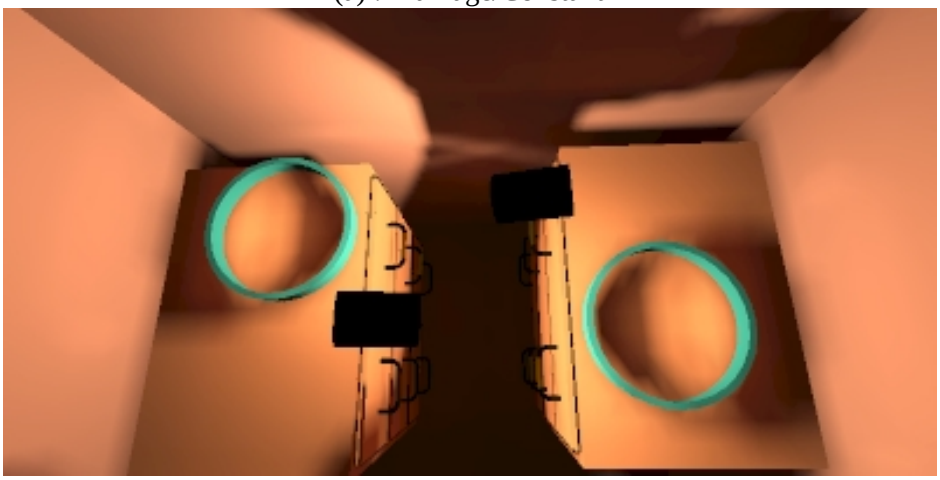
2 Reconstruction des caustiques par textures

La reconstruction des caustiques grâce à une subdivision de la représentation hiérarchique de la radiosité, comme cela a été présenté en première approche (cf. Chap. 4) conduit vite à une limitation dans la complexités des scènes qu'il est possible de traiter. Cette représentation des détails fins (de hautes fréquences en général) nécessite l'emploi de maillages fortement subdivisés. Dans le cadre de scènes réalistes, assez complexes, (comme par exemple pour un bâtiment entier avec du mobilier, tables, chaises, etc), cela implique un coût mémoire inacceptable rendant la simulation impossible. De même, cette forte subdivision entraîne un surcoût dans le calcul de l'éclairage global pour toutes les étapes basées sur un parcours de la hiérarchie. De plus, cela peut créer de fortes différences hiérarchiques entre le calcul diffus et directionnel, visibles dans la reconstruction finale.

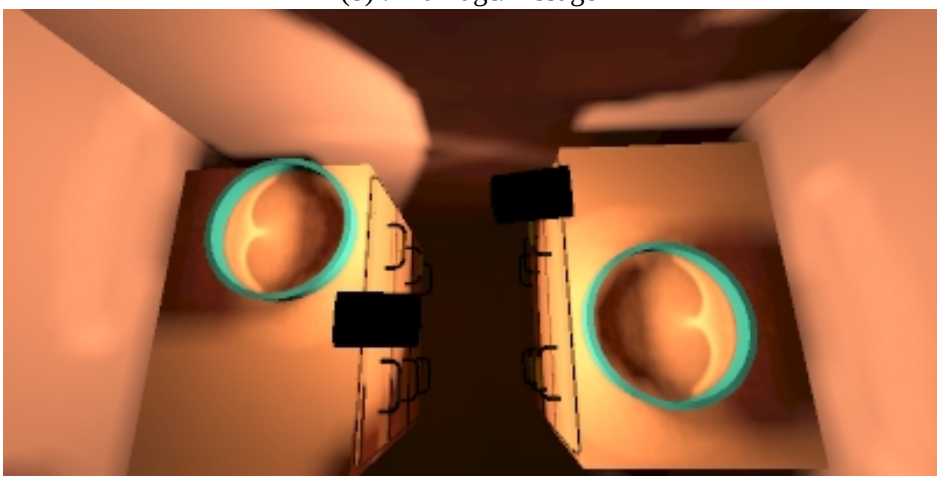
Pour résoudre cet obstacle, nous avons décidé d'utiliser des textures pour représenter les caustiques, lorsque cela est nécessaire. Notre approche est similaire dans



(a) : Maillage/Constant



(b) : Maillage/Lissage



(c) : Texture+Maillage

FIG. 5.2 : Une commode avec des caustiques

l'idée à celle de Heckbert *[Hec90]*. Elle se rapproche aussi de l'utilisation des cartes de photons ("Photon Map") introduits par Wann Jensen *[Jen96]* ; notre principale différence est l'utilisation d'une représentation qui n'est plus indépendante des données, mais fortement liée à chaque objet de la scène. Ceci nous permet d'utiliser directement les capacités de traitement des textures des machines actuelles. Nous présentons aussi des solutions automatiques pour un certain nombre de paramètres dont nous parlerons dans les paragraphes suivants. Il est aussi important de remarquer que notre but est très différent de celui de Wann Jensen, puisque que nous voulons calculer des solutions qui puissent être visualisées interactivement lors d'une balade interactive, par l'intégration des capacités d'un algorithme de radiosité hiérarchique avec regroupement et d'un lancer de particules. De plus, notre solution est adaptée au cas des scènes dynamiques, puisque nous pouvons modifier les textures pendant une interaction avec la scène.

Les trois problèmes suivants doivent être résolus pour permettre l'utilisation des textures pour les caustiques. Le premier est le choix entre les deux types de représentation disponibles : pour des effets lumineux de faibles fréquences spatiales, provenant de la dispersion de l'énergie lumineuse lors de réflexions (comme dans le cas diffus), la représentation des transferts $(L|D)S^*D$ peut être effectuée de manière satisfaisante à l'aide du maillage créé par l'algorithme de radiosité hiérarchique avec regroupement, comme dans la version initiale de l'algorithme unifié (cf. Chap. 4). Nous proposons ainsi un critère permettant de choisir si la représentation la plus adaptée est un maillage ou une texture.

Le deuxième problème, si l'emploi d'une texture a été décidé, est de parvenir à déterminer quelle doit être sa résolution. Pour parvenir à cela, nous utilisons une approche basée sur la réduction de la variance. Enfin, dernier problème, il faut reconstruire la fonction d'éclairage et les textures afin de pouvoir les afficher, et cela, en limitant les calculs aux seules zones nécessaires.

Il est à noter que l'utilisation des textures pour les caustiques n'affecte en rien la solution d'éclairage global qui est toujours calculée par l'algorithme unifié présenté dans le chapitre Chap. 4. Les particules restent stockées et intégrées dans la hiérarchie pour la radiosité comme précédemment. Elles continuent ainsi à apporter leur contribution à la solution globale. Les textures ne sont là que pour améliorer la qualité du rendu tout en réduisant la taille du maillage nécessaire. Réduire le maillage nécessaire permet bien sûr un gain sur la rapidité de calcul, mais aussi sur la consommation mémoire *[GD99]*.

L'utilisation de textures dans ce contexte rend alors possible le traitement de scènes plus complexes comprenant de l'éclairage indirect non négligeable ainsi que des effets directionnels indirects comme des caustiques. La restriction des textures à une zone limitée est essentielle dans ce contexte (cf. Sec. 2.3), améliorant d'autant la qualité des images en limitant le calcul aux zones considérées.

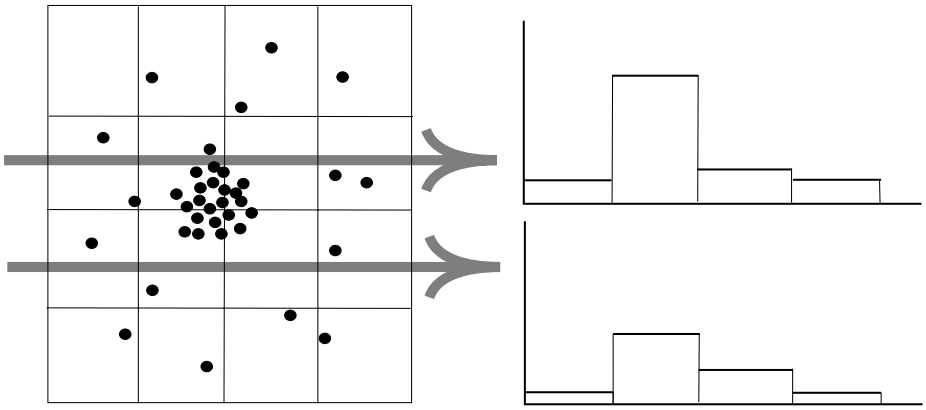


FIG. 5.3 : Construction de l'histogramme

2.1 Choix de la représentation

Intuitivement, l'utilisation d'une texture peut se révéler nécessaire si un grand nombre de particules sont arrivées sur un objet, et si leur répartition n'est pas uniforme.

Après le lancer de particules (cf. Chap. 4 Sec. 3), celles-ci sont accumulées sur les éléments hiérarchiques. Pour déterminer leur nombre sur un objet initial (par exemple les polygones qui n'ont pas été subdivisés), une traversée de la hiérarchie qui en dépend est effectuée. Si ce nombre est inférieur à une certaine valeur[1], une pré-texture ou un histogramme est créé. Cette première évaluation rapide permet de déterminer si l'emploi d'une texture est approprié pour représenter l'influence des particules. Les dimensions de l'histogramme sont choisies par la méthode décrite dans la section 2.2, car celle-ci est aussi utilisée pour calculer la résolution de la texture représentant les caustiques.

Nous calculons ensuite le flux arrivant sur chaque cellule de l'histogramme. Si chaque particule p a une énergie ϕ_p, la cellule $(i,\ j)$ de l'histogramme de dimension n x m contient la valeur (cf. Fig. 5.3) :

$$h(i,j) \ = \ \frac{n \times m}{A_h} \sum_{p \in (i,j)} \phi_p \tag{5.1}$$

où A_h est l'aire de l'histogramme et $p \in (i,j)$ signifie que la position de la particule t est dans la case (i,j). $A_h/(n\ m)$ représente alors l'aire d'une cellule de l'histogramme. Pour un quadrilatère, A_h est égale à l'aire du polygone.

Nous estimons ensuite la répartition uniforme des particules par la valeur :

$$V \ = \ \frac{(t_{min} \ - \ t_{max})}{t_{max}} \tag{5.2}$$

où t_{max} et t_{min} sont les valeurs maximum et minimum présentes dans l'histogramme. Nous utilisons alors le test suivant pour déterminer si une texture doit vraiment être

[1]Elle est en général fixée à 10.

créée :

$$V < \eta \tag{5.3}$$

Ici, η est un paramètre réglable par l'utilisateur[1], représentant le pourcentage de variation acceptable. Si l'inéquation Eq. (5.3) se trouve vérifiée, il n'est pas nécessaire de créer une texture.

Les avantages de l'utilisation des textures pour les caustiques peuvent être perçus immédiatement. En regardant la figure Fig. 5.2, une nette différence est perceptible entre l'image Fig. 5.2-(b) où la caustique est reconstruite en utilisant le maillage existant, et Fig. 5.2-(c) où une texture pour les caustiques a été créée. Pour un même nombre de particules et pour une même qualité de la solution de l'algorithme unifié, nous pouvons remarquer une très forte amélioration de la qualité visuelle.

2.2 Déterminer la résolution de la texture

Le choix de la résolution d'une texture pour une caustique est critique, puisque il a bien sûr une influence sur la qualité visuelle, mais aussi sur la consommation de place mémoire, sur le coût de calcul, et sur la rapidité d'affichage. Ce choix peut être considéré comme partie intégrante de l'étape de reconstruction pour l'affichage. Le principe est simple : une texture de grande résolution nécessite un grand nombre de particules pour obtenir un résultat avec peu de bruit. De même, une texture de faible résolution peut être considérée comme un filtre (un lissage), qui bien sûr diminue le bruit, mais accroît le biais (cf. Chap. 2-Sec. 3.1). En se basant sur cette approche, nous avons décidé de choisir la résolution des textures pour les caustiques en fonction d'un critère basé sur la variance, comme dans le cadre de l'estimation de densité proposée par Walter et al. *[WHSG97]*.

Si la f est la fonction à estimer, la variance est déterminé par :

$$\forall\, x, Var(f(x)) \;\leq\; \frac{\int K^2}{n\,h^2} f(x) \;\leq\; \frac{\int K^2}{n\,h^2} \tag{5.4}$$

où K est le noyau normalisé pour l'estimation de densité, n est le nombre de particules et h est la taille du noyau utilisé (h détermine la finesse de reconstruction).

Puisque nous travaillons sur des textures dont le but est d'être affiché avec OpenGL, elles doivent avoir une taille de $2^d x 2^{d'}$, où d et d' représentent les résolutions de la texture dans les deux dimensions. En première approximation, nous prendrons une texture carré, de taille donc 2^{2d}. Dans ce cas, la taille du noyau de reconstruction est égal à la moitié de la taille d'un pixel :

$$h \;=\; \frac{1}{2}\sqrt{\frac{1}{2^{2d}}}, \tag{5.5}$$

[1] Il est en général fixé à 1%.

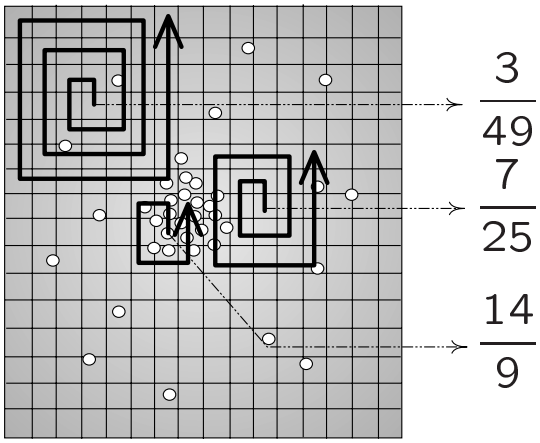


FIG. 5.4 : Trois spirales de reconstruction

Nous choisissons alors la résolution afin que la variance soit inférieure à une certaine précision ε :

$$\frac{K\,2^{2d}}{n} \leq \varepsilon. \tag{5.6}$$

Ainsi, la résolution d est déterminée par l'équation suivante :

$$d \geq \frac{ln(\frac{n\varepsilon}{K})}{2ln2} - 1. \tag{5.7}$$

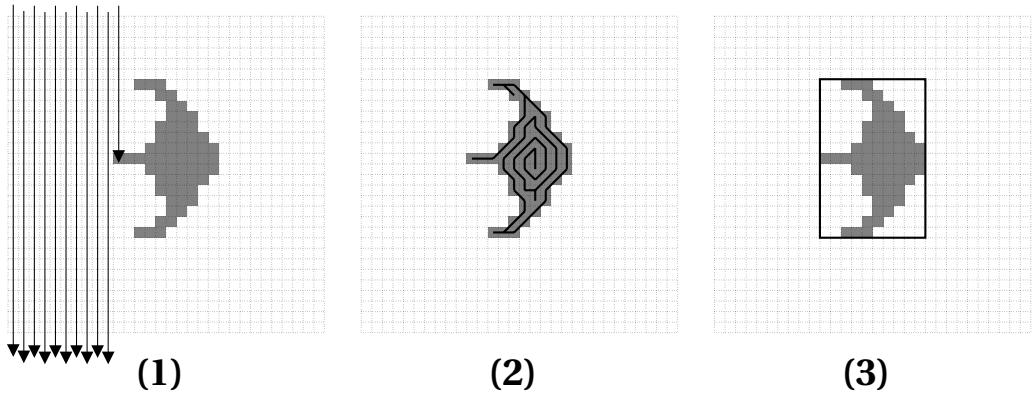
La valeur de K est une constante *[WHSG97]*, correspondant à l'intégrale du noyau choisi sur son domaine. Dans notre cas, nous utilisons un noyau normalisé, donc $K = 1$. La valeur de ε est choisie par l'utilisateur[1] , et constitue en quelque sorte la variance tolérée pour le rendu final. L'influence de ce paramètre est le suivant : la réduction de la tolérance sur la variance va avoir tendance à trop lisser la reconstruction, tandis qu'une tolérance plus grande va augmenter le bruit.

2.3 Filtrage

Lorsque le choix de la création d'une texture pour les caustiques a été fait, nous déterminons sa résolution comme décrit précédemment. Puis nous passons par une étape de filtrage pour réduire le bruit et ainsi améliorer la qualité de la texture qui sera utilisée pour l'affichage. Encore une fois ici, cette étape constitue un compromis entre le bruit et le biais. Nous utilisons pour cela un noyau de lissage similaire dans l'esprit à celui utilisé par Wann Jensen *[Jen96]* dans le "Photon Map".

La première étape consiste à trouver la valeur maximale t_{max} de l'histogramme. Ensuite, pour chaque pixel, nous faisons alors un parcours en spirale (cf. Fig. 5.4) en

[1]Elle est en général fixée à 1.



(1) Recherche de la caustique (2) Parcours de la caustique (3) Boite englobante

FIG. 5.5 : Détection de la zone de restriction

sommant les valeurs dans une autre texture t', comme décrit ci-dessous :

$$t'(i,j) \;=\; \frac{\sum_{r\leq R} t(i,j)}{N} \tag{5.8}$$

Ici, R est le rayon maximal autorisé par l'utilisateur, r est le rayon courant, et N est le nombre de pixels correspondant au voisinage défini par le noyau. De plus, le parcours s'arrête dès que le rayon $r' \leq R$ est atteint,r' vérifiant :

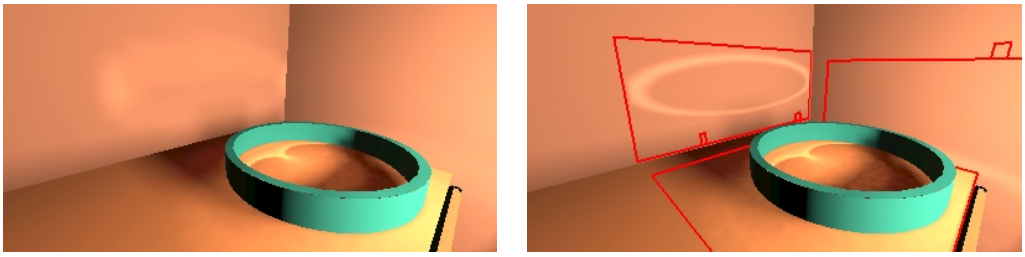
$$\sum_{r\leq r'} t(i,j) \;\geq\; t_{max} \tag{5.9}$$

Ainsi, à l'aide de ces critères, le noyau de filtrage s'adapte aux variations locales de densité de particules. Cela permet de conserver les fortes variations lorsqu'elles existent, mais de bien lisser là où il n'y a que très peu d'information. Les zones de faible densité sont très lissées, alors que celle de forte densité proche de t_{max} restent à leur valeur d'origine.

2.4 Restriction

L'utilisation d'une texture couvrant la totalité d'une surface comme un polygone peut se révéler inefficace. À travers les images de la figure Fig. 5.6, nous pouvons remarquer que seule des régions restreintes du mur et du dessus de la commode sont éclairées par les réflexions directionnelles des lampes sur l'anneau spéculaire. Une partie de la reconstruction se fait donc dans des zones qui ne sont que peu concernées par la présence d'une caustique. C'est pourquoi, il peut être avantageux de restreindre les textures à des zones plus localisées, réellement affectées par les réflexions directionnelles.

Pour parvenir à cela, nous utilisons la valeur maximum de l'histogramme t_{max} présenté ci-dessus. Nous parcourons alors l'histogramme jusqu'à la découverte d'un



Sans restriction Avec restriction

Les régions de restrictions sont marquées en rouge.

FIG. 5.6 : Étude comparative de la restriction

pixel ayant une valeur suffisante, c'est à dire qu'elle est supérieure à un certain pourcentage (en général entre 1 et 5%) de la valeur maximale t_{max}. À partir de cette position, nous continuons le parcours de proche en proche uniquement sur les pixels répondant à ce même critère. En procédant ainsi, nous nous déplaçons à l'intérieur de la caustique et nous en déterminons sa forme. Celle-ci va nous permettre de calculer les coordonnées de texture minimales et maximales à l'intérieur de laquelle une caustique existe vraiment. Nous les utilisons pour construire une boîte englobante, et pour ainsi reconstruire une texture limitée à cette région. S'il existe plusieurs zones sur la surface, nous recommençons le processus à partir de la dernière position, après avoir mis à zéro tous les pixels de la zone concernée par la texture précédente.

Le résultat de la restriction est évidemment de plus grande qualité, car nous utilisons une résolution de même ordre de grandeur sur une zone restreinte. Cela peut être vu dans la figure Fig. 5.6 à droite, sur le mur.

3 Lancer de particules incrémental

Un des buts principaux de cette nouvelle approche est de pouvoir limiter le nombre de particules à émettre nécessaires lorsqu'un objet se déplace, ceci afin d'accélérer la mise à jour. Dans l'algorithme unifié précédant (voir Chap. 4), toutes les particules sont réémises lors du déplacement d'un objet. Bien que l'efficacité de cet algorithme permette une certaine interactivité pour des scènes de faible complexité, cela devient vite impossible même pour des environnements modérément complexes. Pour parvenir à des mises à jour plus rapides, voire interactives, nous basons notre approche sur la hiérarchie des liens afin de traiter efficacement le cas diffus, et sur une restriction du nombre de particules qui sont réémises pendant le mouvement de l'objet.

Comme pour le cas diffus, les effets lumineux directs du mouvement de l'objet sur les transferts directionnels sont limités dans l'espace, mais sur de plus longs chemins, à cause des multiples réflexions directionnelles. L'idée directrice de l'algorithme incrémental que nous proposons est de trouver tous les chemins lumineux qui sont affectés par le déplacement, et de réémettre les particules correspondant à ces chemins.

Nous avons choisi d'utiliser une structure spatiale adaptative (sous la forme d'un octree), ceci pour permettre une détection efficace. Chaque case de l'octree conserve les références vers des liens à partir desquels il est possible de reconstruire les chemins lumineux.

3.1 Octree dynamique

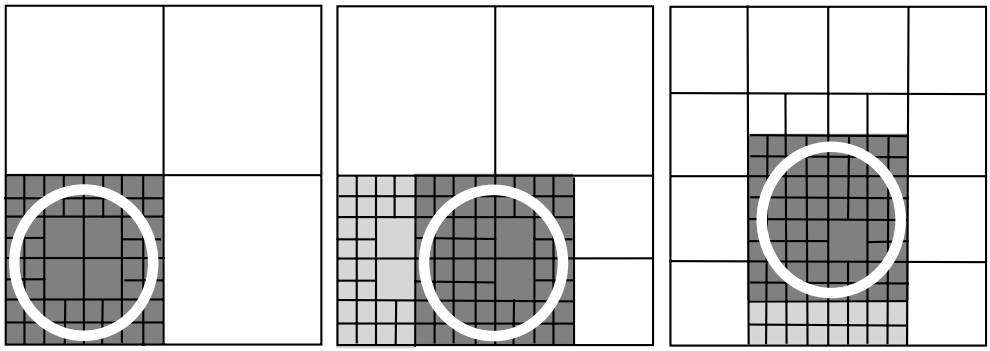
Avant tout mouvement, notre octree est vide. Lorsqu'un objet est sélectionné pour un mouvement ("objet dynamique"), nous subdivisons l'octree autour de celui-ci jusqu'à une profondeur maximale définie par l'utilisateur (cf. Fig. 5.7-(1)). A chaque mouvement de l'objet l'octree est adapté à sa nouvelle position. Trois types de cellules sont définies dans l'octree : celles contenant l'objet, celles qui contenaient l'objet avant le mouvement, et celles qui n'ont pas été affectées par le déplacement (cf. Fig. 5.7-(2-3)). Les cellules qui ne sont pas affectées sont alors simplifiées pour limiter le coût de stockage. La figure Fig. 5.8 en montre un exemple

Lors de la propagation d'une particule, le lien dont elle est issue est inséré dans les cellules de l'octree intersectées par son chemin lumineux. Chacune de ces cellules maintient donc l'ensemble des liens qui peuvent avoir une interaction avec la zone spatiale qu'elle définit. Cet ensemble est représenté sous la forme d'une table de hachage indexée sur l'élément récepteur r sur lequel le lien l est stocké. Évidemment, les seuls liens à considérer sont ceux qui permettent un transfert diffus vers directionnel (DS^*), c'est à dire ceux dont la propriété de réflexion du récepteur contient une composante directionnelle. Les liens directement en interaction avec l'objet dynamique peuvent aussi être exclus et traités séparément. L'octree est là pour détecter les influences indirectes.

Par une telle structure, il est facile de détecter quels sont les liens dont l'émission de particules est affectée par le mouvement de l'objet dynamique. Ce sont ceux qui se situe dans la zone où se trouve (et se trouvait à l'instant d'avant) l'objet dynamique. De plus, lorsque nous accédons à la table de hachage pour identifier les liens, le fait qu'elle soit indexée par le récepteur permet de retrouver facilement l'élément affecté, sur lequel une nouvelle émission va devoir être effectuée.

3.2 Mise à jour incrémental des particules

Pour permette la réémission des particules nécessaires, nous récoltons l'ensemble des liens dans chacune des cellules de l'octree affectées par le mouvement. Il faut aussi considérer celles en interaction directe avec le mouvement. Puis, pour chacun de ces liens, il faut enlever toutes les particules qui ont été émises à travers lui, ainsi que leurs contributions, et ceci, sur chacun des éléments affectés par leur propagation. Pour faciliter cette suppression, les impacts des particules sont chaînés à partir du lien émetteur et chaque impact conserve en plus l'information de l'élément qui le supporte. Ainsi, l'influence de la particule est supprimée sur tous les éléments qui ont été affectés



Initialisation (première insertion de l'objet cylindrique) (0) puis mouvement vers la droite (1) vers le haut (2). En gris foncé, les cellules considérant l'objet comme présent, en gris clair, l'objet était présent avant le mouvement.

FIG. 5.7 : Description de l'octree dynamique

par sa réflexion ou sa réfraction. Ensuite, des particules sont réémises à partir de ces liens.

Tous les éléments affectés par cette mise à jour des particules, sont marqués comme ayant subis une modification. Ainsi, comme pour le cas diffus avec l'utilisation de la hiérarchie des liens (cf. Sec. 1.2 et Sec. 4), la mise à jour hiérarchique (Push/Pull) peut être faite partiellement tout en conservant la représentation multirésolution.

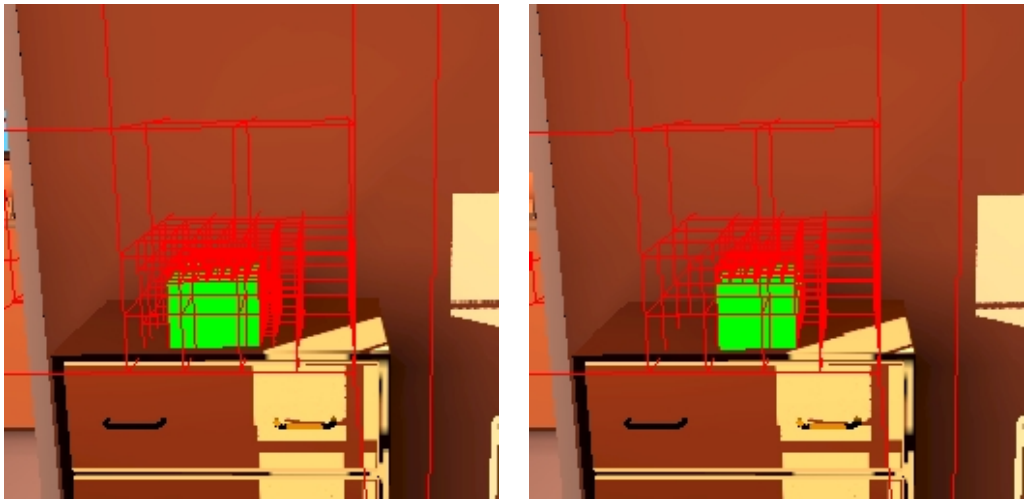
Il faut noter que pour obtenir des effets secondaires dus aux transferts DS^*D, il suffit de répéter plusieurs fois l'étape de transfert d'énergie et de mise à jour hiérarchique (cf Chap. 4).

3.3 Mise à jour interactive avec dégradation de qualité

Malgré un facteur d'accélération assez élevé comparé au recalcul de la solution complète, la vitesse de mise à jour de l'algorithme décri ici reste assez lente, avec un résultat d'environ 20-30, par exemple, secondes par image pour la scène d'intérieur (cf. Fig. 5.11). Pour arriver à une vitesse quasi-interactive, il est possible de limiter le nombre de particules à émettre durant le mouvement, mais avec bien sûr une perte de qualité. C'est un compromis au profit de la rapidité.

Pendant le mouvement, nous mémorisons les liens affectés, donc ceux à travers lesquels nous devons réémettre des particules. De plus, un faible pourcentage (paramétrable par l'utilisateur) de particules est réémis par rapport aux cas sans dégradation de qualité. Cela permet une accélération notable de la mise à jour, de 30 secondes par image à 3 secondes par image sur la scène d'intérieur (cf. Fig. 5.9) [1].

[1] Tous les résultats sont calculé sur SGI Origin 2000 avec un processeur MIPS à 250Mhz



Position initiale Après le mouvement

Remarquez la simplification de l'octree par rapport à la position précédente de l'objet.

FIG. 5.8 : Le mouvement capté par un octree

Lorsque le mouvement s'arrête, il suffit simplement de réémettre toutes les particules sur tous les liens sauvegardés, afin d'obtenir une image de plus grande qualité (cf. Fig. 5.9). Cette approche permet à l'utilisateur d'avoir un meilleur contrôle sur l'interactivité de la mise à jour, en jouant sur le compromis rapidité de calcul et la qualité de la solution.

4 Amélioration de la solution incrémentale pour le cas diffuse

Comme nous l'avons déjà mentionné dans la section Sec.1.2, nous utilisons une hiérarchie des liens comme dans *[DS97]* pour permettre une mise à jour incrémentale et rapide pour la solution diffuse. Cependant, nous avons apporté quelques améliorations à cette approche.

Une des limitations de la méthode originelle consistait en la nécessité de calculer et *de conserver les volumes d'interaction* ("shafts") pour la hiérarchie de liens en entier. Ce problème a aussi été identifié par Schoeffel et Pomi *[SP99]*, qui préconisait alors l'emploi d'une approche prédictive pour limiter ces calculs. Dans notre cas, nous avons décidé de n'en conserver aucun, et de ne les recalculer que lorsque cela est nécessaire. Cela n'apporte pas un gros surcoût de calcul à condition qu'une boîte englobante soit stockée sur tous les éléments de la hiérarchie. En effet, chaque volume d'interaction est consulté au plus une fois par lien, par mise à jour et par objet dynamique. De plus, la seule information des boîtes englobantes permet de résoudre un certain nombre de cas sans avoir à construire les plans du volume d'interaction. Les étapes du calcul

d'intersection entre une boîte englobante b et un volume d'interaction entre les boîtes b_s et b_r sont alors les suivantes :

1. Si b est à l'extérieur de la boîte englobante de b_s et de b_r, alors b est en dehors du volume d'interaction.
2. Sinon si b est à l'intérieur de b_s ou de b_r, alors b est à l'intérieur.
3. Sinon si b intersecte b_s ou b_r, alors b intersecte le volume d'interaction.
4. Sinon, tout en construisant les plans du volume d'interaction, nous testons si b est à l'extérieur.
5. Si le test précédant a échoué, il reste à tester avec les plans nouvellement créés si b est à l'intérieur.

Ce n'est que dans le pire des cas (les 5 tests doivent être effectués) que tous les plans du volume d'interaction doivent être construits. Le pseudo-code de cette procédure d'intersection est disponible dans l'annexe An. B. La mémoire économisée est significative, puisque le stockage de tous les volumes d'interactions nécessitait une surcoût mémoire prohibitif pour l'application à des scènes complexes (cf. *[SP99]* pour l'analyse de l'utilisation mémoire de l'algorithme originel).

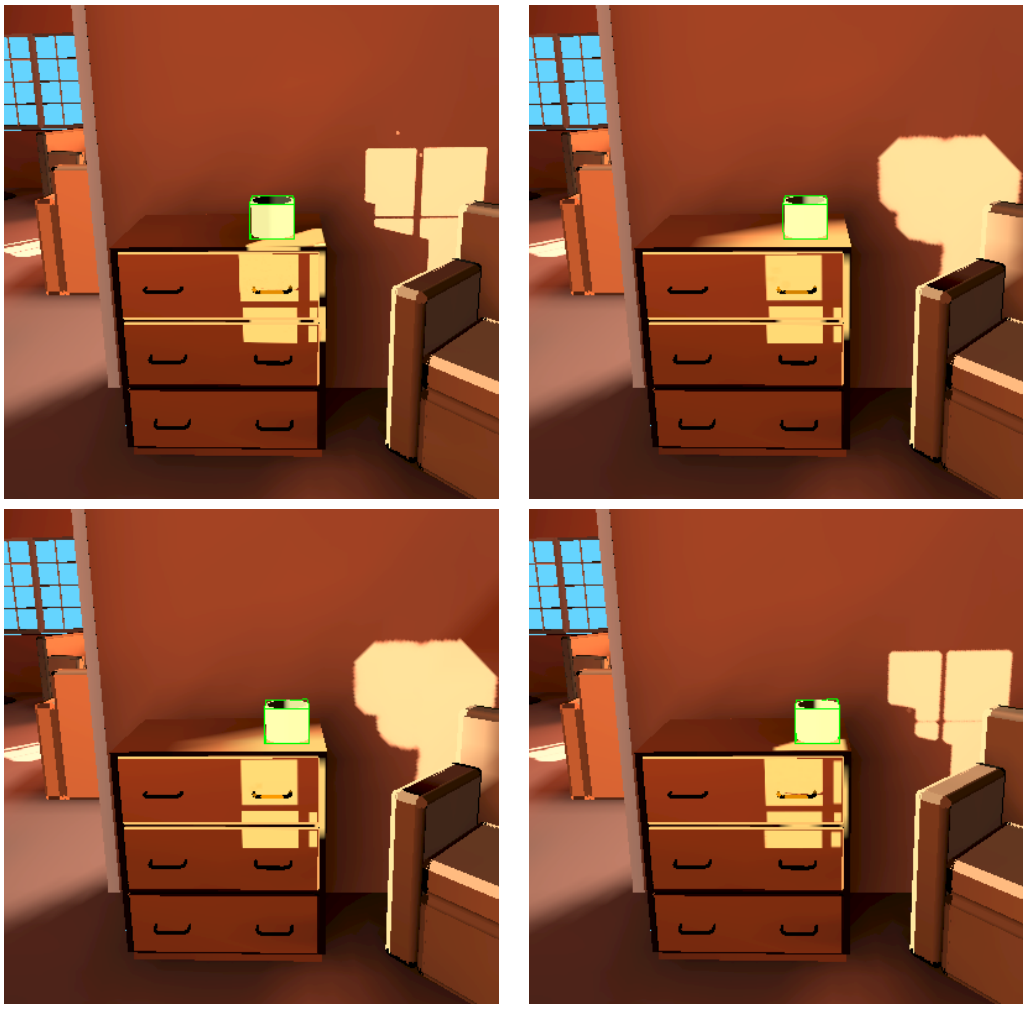
Nous utilisons aussi la hiérarchie des liens, telle que nous l'avons présentée dans *[GD99]*, qui offre une alternative à la numérotation de la méthode d'origine *[DS97]*. La seule information gardée sur le lien est la méthode qui a permis son raffinement en plusieurs autres liens : les liens fils ont-ils été obtenus par la subdivision du récepteur ou de l'émetteur ? Cette information nous permet ainsi de retrouver facilement les liens fils.

5 Affichage interactif de la solution

Nous développerons ici les techniques qui nous ont permis d'arriver à afficher rapidement la solution mise à jour. Le calcul doit être rapide, mais il faut aussi pouvoir le visualiser rapidement. Il faut ici noter que pour accélérer le rendu de la solution incrémentale, seuls les objets visibles à l'écran seront affichés. De même, c'est seulement sur ces objets que les reconstructions seront effectuées.

5.1 Affichage des textures de caustique

Pour cumuler les effets représentés dans les textures représentant les caustiques avec la solution diffuse comprise dans le maillage lors de l'affichage, chaque objet texturé est rendu en deux étapes. Tout d'abord, il faut remarquer que les textures représentant les caustiques ne sont rendues que sur les objets de plus haut niveau, comme un polygone initial, non-subdivisé. Pour un tel polygone, nous désactivons en premier l'écriture dans le tampon de Z (tout en conservant le test) et pour chaque texture représentant des caustiques, nous rendons le polygone texturé. Nous activons



(a) Position initiale avec des ombres de bonne qualité. (b-c) Deux positions intermédiaires durant un déplacement à 3-4 secondes par image. La qualité est ici dégradée. (d) Fin du déplacement, 20 secondes après, nous obtenons une image de bonne qualité.

FIG. 5.9 : Déplacement du pot sur la commode

ensuite l'écriture dans le tampon de Z, afin de rendre le maillage avec l'éclairage de l'objet considéré. L'éclairage présent sur le maillage est sommé avec les textures en utilisant la fonction de mélange OpenGL `GL_ADD`.

```
glDepthMask(GL_FALSE)
glEnable(GL_TEXTURE_2D)
pour chacune des textures T :
   rendre le polygone avec la texture
glBlendFunc(GL_ONE,  GL_ONE)
glDisable(GL_TEXTURE_2D)
```

```
glDepthMask(GL_TRUE)
rendre le maillage
glBlendFunc(GL_ONE,  GL_ZERO)
```

Cette méthode permet d'afficher une nombre illimité de textures par polygone en une seule passe.

Pour les textures restreintes, il ne faut pas les répéter sur un objet. Pour cela, nous utilisons le paramètre `GL_CLAMP` et une bordure noire (valeur nulle) pour éviter la propagation du dernier pixel dans ce mode d'affichage de texture.

5.2 Paramétrisations des textures de caustique

Afin de pouvoir créer les textures pour les caustiques, il faut savoir transformer les impacts 3D des particules en des coordonnées de texture 2D. Pour cela, nous utilisons une paramétrisation 2D $[0,1]x[0,1]$ implicite définie pour chaque type d'objet de notre système (polygone, sphère, cône, cylindre, cf. Chap. 3).

Pour les textures restreintes, nous calculons une transformation matricielle M_r définissant une correspondance entre la paramétrisation implicite et la paramétrisation restreinte. Si $c_i = (x_i, y_i, 1)$ est la paramétrisation implicite calculée à partir de la position 3D, la position dans une texture restreinte est $c_r = (x_r, y_r, 1) = M\, c_i$, M étant définie par

$$M = \begin{bmatrix} \alpha(y_2 - y_1) & \alpha(x_1 - x_2) & \alpha(x_0(y_1 - y_2) + y_0(x_2 - x_1)) \\ \alpha(y_0 - y_1) & \alpha(x_1 - x_0) & \alpha(x_0 y_1 - y_0 x_1) \\ 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

avec (x_0, y_0), (x_1, y_1), (x_2, y_2) les coordonnées dans la paramétrisation implicite des positions $(0,0)$, $(1,0)$ et $(1,1)$ de la texture restreinte, et avec

$$\alpha = 1/[\, x_1 * y_2 - x_0 * y_2 + x_0 * y_1 - x_2 * y_1 + x_2 * y_0 - y_0 * x_1 \,] \tag{5.11}$$

De même

$$M^{-1} = \begin{bmatrix} x_1 - x_0 & x_2 - x_1 & x_0 \\ y_1 - y_0 & y_2 - y_1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.12}$$

Avec une telle transformation, le calcul de la position d'une particule dans une texture restreinte est facilité. Il suffit de trouver la position implicite et de lui appliquer la transformation définie ci-dessus.

6 Résultats

Nous avons testé notre approche sur deux scènes assez différentes, une scène d'intérieur combinant l'éclairage du soleil et de lampes intérieures, et une scène d'extérieur

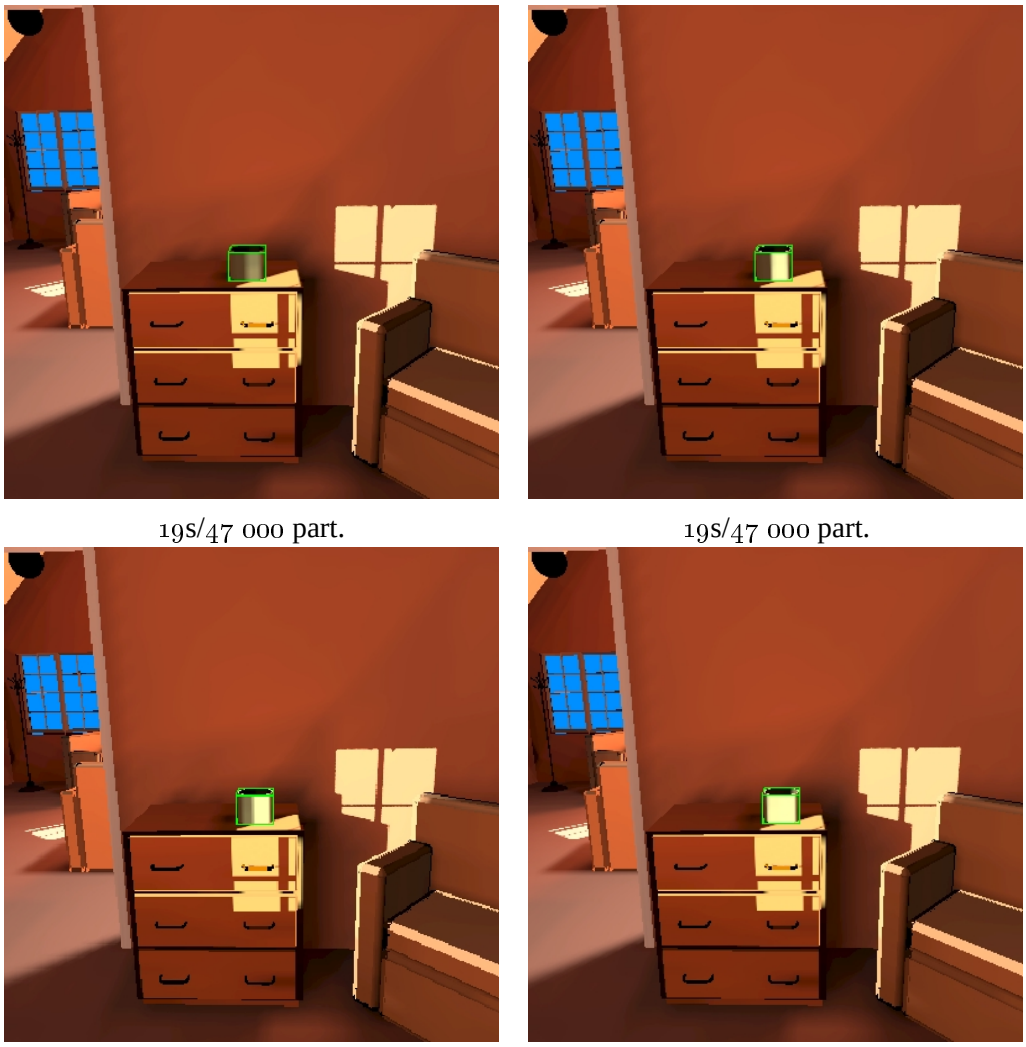
(a) Chambre, avec des caustiques sur les deux commodes créées par les spots (en bas à droite), et des caustiques créées par la réflexion du soleil à travers la fenêtre (en haut) sur le miroir (à droite) sur le mur et la commode (en bas). (b) Salle à manger, réflexion du soleil sur un miroir (en bas à droite, non visible ici). (c) Salle de bain, caustiques créés par le miroir de l'armoire sur le mur à gauche. Toutes les images sont rendues grâce à OpenGL et le "Render Cache" *[WDP99]* pour la composante dépendante du point de vue.

FIG. 5.10 : Scène d'intérieur

dans laquelle l'éclairage indirect directionnel fournit une grande partie de l'intensité lumineuse. Les tests ont été mené sur une SGI Origin 200 (processeur MIPS à 250 MHz).

La scène d'intérieur est une maison complète avec trois pièces [1] : une chambre, une salle de bain, et une salle à manger (cf. Fig. 5.10). Dans un algorithme de radio-

[1] la scène originelle est disponible sur le site `http ://floyd.lbl.gov/mgf/scenes.html`.



19s/47 000 part. 19s/47 000 part.

20s/47 000 part. 22s/57 000 part.

Le temps indiqué est le temps de mise à jour entre deux images. Remarquez l'accroissement de l'intensité lumineuse sur le récipient et le déplacement de l'ombre par derrière.

FIG. 5.11 : Quatre postions du récipient sur la commode de la chambre

sité classique, si les objets de haut niveau sont subdivisés jusqu'à une profondeur de 5, la scène contiendrait 60 000 polygones. Elle ne comporte pour nous que environ 4 000 objets initiaux. La solution d'éclairage global comprend plusieurs effets lumineux directionnels dus aux réflexions des lampes présentes dans les pièces, mais aussi à l'influence d'un soleil levant éclairant les pièces à travers les fenêtres. Des images des trois pièces sont disponibles dans la figure Fig. 5.10. Elles montrent la présence de plusieurs réflexions spéculaires et de plusieurs caustiques.

Pour obtenir la convergence de la solution (quand la variation de l'énergie moyenne est suffisamment faible), notre algorithme nécessite 6 itérations, pour un temps total

de 35 minutes et 44 secondes Le nombre total de particules qui sont envoyées à chaque itération est en moyenne de 1,6 millions. Nous utilisons 25 textures restreintes pour les caustiques, avec une résolution moyenne de 85*x*85 pixels.

Pour la mise à jour incrémentale, nous avons déplacé un récipient cylindrique sur le dessus d'un commode dans la chambre, afin de le positionner dans les reflets du soleil à travers le miroir. Quatre images correspondant à quatre positions du récipient sont disponible dans la figure Fig. 5.11. Les temps de calcul et le nombre total de particules renvoyées y sont aussi décrits.

Comme vous pouvez le remarquer, le nombre de particules renvoyé est situé aux environs de 3-4% du nombre total nécessaire pour le calcul complet de l'éclairage de la scène. Il en résulte une forte augmentation de la vitesse de calcul, comparé au temps nécessaire pour la solution initiale. Ainsi, il est possible de déplacer un objet en environ 20 secondes par image, par rapport aux 35 minutes nécessaires pour tout recalculer.

Plus de détails se trouvent dans le tableau Tab. 5.1. Le surcoût mémoire nécessaire pour stocker l'octree est de 370Ko. Pour une moyenne d'environ 3% de particules réémises, la mémoire consommée par cette structure reste très modeste.

Scène	TIC	NTP	MO
Intérieure	35mn44s	1 600 000	370 Ko
Extérieure	1h19mn	800 000	2,3 Mo

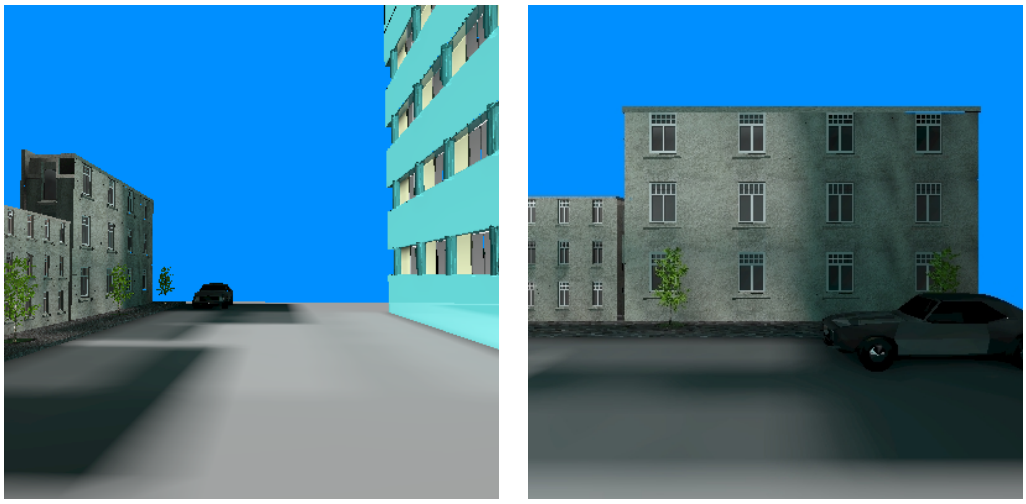
Scène	MT	MP	PPT
Intérieure	14,5s	0-97 000	0-6%
Extérieure	96s	5 700-41 000	0.8-17%

TIC : Temps initial de convergence. NTP : Nombre total de particules, MO : mémoire utilisé par l'octree. MT : moyenne du temps de mise à jour dans le déplacement d'un objet (récipient/voiture). MP : nombre de particules émises pour une mise à jour, PPT : pourcentage du nombre total de particules.

TAB. 5.1 : Résultats de deux sessions incrémentales

Le second exemple est une scène d'extérieur. Un aperçu général est disponible dans la figure Fig. 5.12. La lumière du soleil provient de la gauche, et se réfléchit sur l'immeuble à droite, éclairant ainsi les bâtiments à gauche ainsi que la route. Une voiture se déplace entre les deux bâtiments, projetant une ombre visible sur la gauche des images. La scène contient 47 000 polygones initiaux, et 118 000 après le calcul complet de la solution d'éclairage global.

Pour obtenir une solution qui converge, il a fallu 1 heure et 19 minutes, avec en moyenne 800 000 particules à chaque itération. Le temps de mise à jour moyen est d'une minute trente. Dans la figure Fig. 5.13, nous pouvons voir quatre positions de la voiture lors de son déplacement, ainsi que le mouvement de l'ombre de la voiture qui en résulte. Le nombre de particules renvoyées est aussi disponible dans cette figure. Remarquez que dans ce cas, la mise à jour diffuse est dominante dans le temps de mise à jour, à cause du grand nombre de liens créés par notre oracle de raffinement.



La lumière du soleil (source directionnel) vient de la gauche et se réfléchit sur les murs spéculaires de l'immeuble à droite.

FIG. 5.12 : Scène d'extérieur ($47\,000$ polygones)

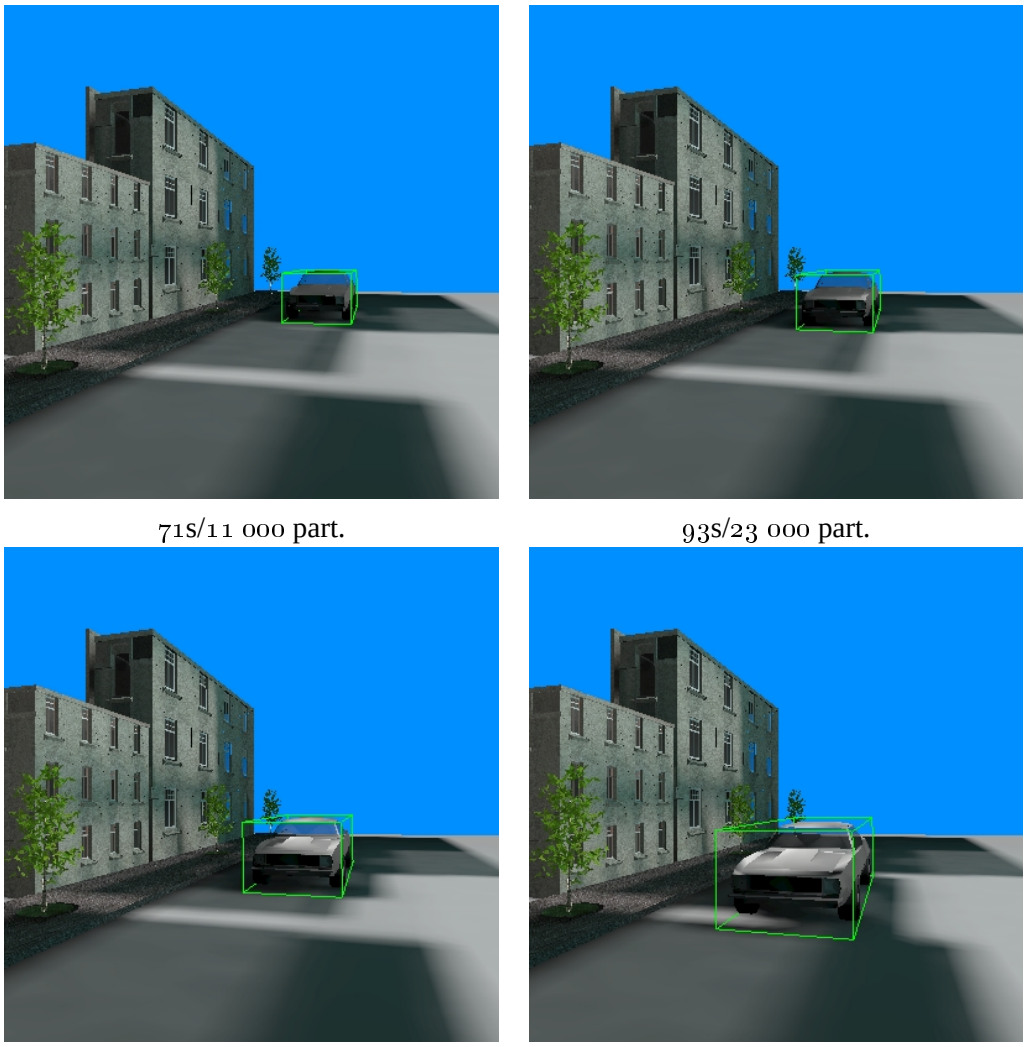
Il est important de remarquer que toutes les images présentées ici sont rendues en utilisant l'accélération graphique matérielle à l'aide de la librairie OpenGL pour la composante indépendante du point de vue, combinée avec le "Render Cache" pour la composante dépendante du point de vue. Ainsi, il est possible de se promener interactivement dans la solution (cf. Chap. 4).

7 Conclusion

Nous avons présenté dans ce chapitre une nouvelle approche pour le traitemement de scènes complexes dans le cadre d'une mise à jour incrémentale. Basée sur l'algorithme unifié du chapitre précédent, elle apporte plusieurs solutions : l'utilisation des textures pour la représentation des détails fins et une approche permettant une mise à jour incrémentale rapide. Ces deux approches nous permettent de traiter des scènes plus complexes tout en autorisant une interaction de l'utilisateur sur la solution afin de contrôler le résultat final.

Nous avons donc démontré dans un premier temps la possibilité de séparer la reconstruction des phénomènes avec de grandes variations comme les caustiques, de la reconstruction de variations plus faibles comme les effets diffus. La méthode décrite permet de choisir automatiquement la méthode de reconstruction qui se trouve la plus adaptée aux variations. Le choix est effectué entre l'usage d'une texture ou du maillage. La résolution des textures est aussi déterminée de manière automatique.

Puis nous avons présenté une nouvelle approche permettant de réduire le nombre de particules à renvoyer lors d'une mise à jour incrémentale pour le déplacement d'un objet. Nous utilisons une structure efficace basée sur un octree. Elle permet de



71s/11 000 part. 93s/23 000 part.

136s/139 000 part. 114s/24 000 part.

Les temps de mise à jour et le nombre de particules utilisé sont indiqués sous chaque image. Remarquez l'accroissement de la luminosité de la voiture et le déplacement de l'ombre de la voiture sur le sol à gauche.

FIG. 5.13 : Quatre positions de la voiture dans la scène d'extérieur

référencer les liens qui ont généré des particules, afin de les identifier rapidement lors d'un déplacement. Il est ainsi possible de mettre à jour très rapidement une solution d'éclairage global (par exemple, en 20 secondes pour une solution qui a nécessité 35 minutes pour obtenir la convergence).

Pour finir, nous avons développé une méthode quasi-interactive avec une dégradation de la qualité, mais qui permet d'avoir une mise à jour bien plus rapide. Cette méthode permet de positionner rapidement des objets tout en mettant à jour l'ensemble des effets lumineux, y compris les caustiques, à un taux de rafraîchissement d'environ

3-4 secondes par image, encore une fois pour une solution qui a nécessité 35 minutes de calcul.

Cependant, bien que les valeurs fixées par défaut donnent les résultats souhaités sur la plupart des scènes testés, de nombreux paramètres ont été introduits. C'est une contrepartie de l'interactivité que nous avons recherchée.

De plus, l'augmentation de la mémoire pour les textures et le support toujours croissant du plaquage de multiples textures ("multitexturing") par les cartes graphiques bon-marché, sont des arguments en faveur de l'utilisation des telles représentation. Il serait possible d'y ajouter la représentation des ombres dans l'esprit de Myszkowski et Kunii *[MK94]* comme de Soler et Sillion *[SS00]* ou tout autre méthode génératrice de textures d'ombres.

Enfin, il faut encore fournir à l'utilisateur la possibilité de générer des images de très grande qualité. Grâce à l'emploi de textures, nous avons ici obtenu un affiche de détails de bonne qualité, comme des caustiques, dans un cadre de visualisation interactive. La solution développée fournit une solution globale, mais il faut pouvoir affiner les résultats pour obtenir une image de très grande qualité à partir des informations déjà présentes. Dans le chapitre suivant nous développerons une première solution permettant l'obtention de telles images dans un cadre dépendant du point de vue.

6

Première approche de la reconstruction finale

Les solutions que nous avons proposées dans les chapitre précédents nous permettent de calculer des solutions d'éclairage globale de bonne qualité pour un affichage indépendant du point de vue. Mais, pour certaines applications, cette qualité n'est pas toujours suffisante. Dans la production de séquences pour un film, nécessite par exemple des images d'une qualité irréprochable, ce qui n'est pas le cas lors de la modélisation ou pour une application plus liée à la simulation ou la réalité virtuelle ou la réalité virtuelle.

Dans ce chapitre, nous allons nous pencher sur la reconstruction finale pour notre algorithme unifié, dans le cadre de calcul d'une image, donc dans une approche dépendante du point de vue. Dans ce cadre, un calcul pour chacun des pixels d'un image est effectué. Cela rend possible une plus grande précision de traitement que pour le calcul d'une solution complète sur l'ensemble d'une scène discrétisée. Il est possible ainsi d'obtenir des images de très grande qualité. Nous présenterons ici des premières solutions et des pistes pour le traitement de cette étape de reconstruction pour notre algorithme unifié. Pour cela, nous ferons largement appel aux notations des chemins lumineux introduit par Heckbert *[Hec90]* (cf. Chap. 2-Sec. 1.4).

Les solutions présentées ici sont des prospectives pour le problème de la reconstruction finale. Elles n'ont pas été toutes complètement implémentées. Nous préciserons dans la partie résultats les approches que nous avons testées.

1 Présentation générale

Un des but de tout algorithme d'éclairage global est de pouvoir fournir une image de haute qualité. Si la méthode unifiée présentée dans les chapitres précédents permet d'obtenir des solutions globales indépendantes du point de vue de bonne qualité, elle fournit aussi un grand nombre d'informations qui peuvent être utilisées dans un cadre dépendant du point de vue, conduisant ainsi vers l'obtention d'une image de grande qualité.

Si maintenant nous nous plaçons dans un cadre dépendant du point de vue, nous pouvons baser notre algorithme sur un tracé de rayons, collectant les informations existantes dans la scène afin de réduire les calculs nécessaires. De plus, un tracé de rayons permet de limiter les effets de crénellage par un sur-échantillonage adaptatif *[PS89]*.

Ainsi, pour chaque rayon émis à partir du point de vue à travers l'image, nous récupérons dans un premier temps le premier objet intersecté, feuille de la hiérarchie due à notre algorithme unifié. Sur cette objet, toute l'information disponible est récupéré, comme par exemple les valeurs de radiosité, les particules et les liens contenant des informations de visibilité et de facteur de forme, etc. Il faut noter que cette action n'est nécessaire qu'une seule fois par élément visible dans l'image. Toute cette information va permettre de calculer la valeur de radiance le long de ce rayon.

La structure générale de notre algorithme est présenté dans la figure 6.1. Une telle architecture permettrait une gestion de type cache pour les informations nécessaires, afin de gérer au mieux la mémoire. En effet, seule les informations locales sont vraiment nécessaires pour le calcul de la valeur d'un pixel. Mais sauvegarder ces informations évite un surcoût de calcul. Un cache sur les informations permettrait un tel compromis.

Puisque la composante indépendante de la direction dans la solution d'éclairage est déjà présente, cette valeur sera simplement retirée de l'information existante, et même améliorée quand ceci est nécessaire. Cela concerne la valeur de radiosité LD^*, comme celle due aux effets spéculaires $(L|D)S^+D$. Il est à noter que la propagation du rayon s'arrête ici pour le calcul de ces quantités.

Il y a deux types d'information dans l'algorithme unifié que nous avons présenté. La première est celle fournie par les liens, et donne des indications principalement sur les *échanges diffus* $(D|L)D$. Les particules, elles, contiennent les informations sur les échanges $(D|L)S^+$. Ces dernières vont donc nous aider à reconstruire la partie indépendante du point de vue $(L|D)S^+DE$, mais aussi les reflets dépendant du point de vue $(L|D)S^+E$.

2 Reconstruction de la composante diffuse

La première étape consiste en le calcul de la valeur diffuse pour le point de l'objet intersecté. Notre méthode va pour cela se baser principalement sur les travaux de

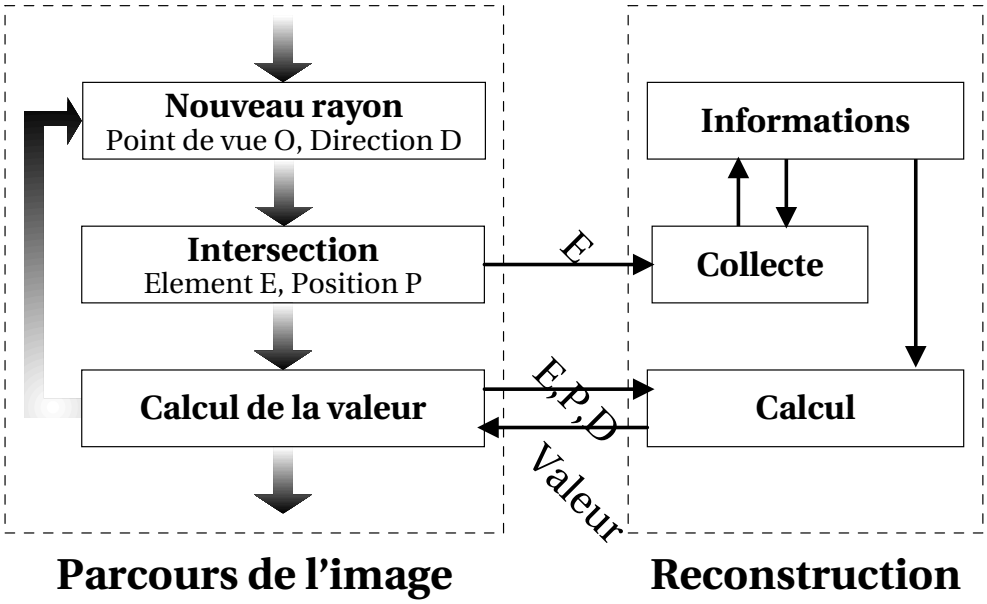


FIG. 6.1 : Aperçu de l'algorithme de reconstruction final

Scheel et al. *[SSS01]*, qui ont présenté une approche globale au problème de la reconstruction finale, mais exclusivement pour des scènes diffuses. Celle-ci peut se révéler coûteuse pour les scènes complexes, puisqu'il faut récolter toute l'information sur chacun des points de maillage de l'environnement. Nous proposons donc une adaptation tirant encore plus partie du fait que le calcul est dépendant du point de vue. Comme cela a été présenté dans le paragraphe précédant, il y a deux étapes principales, la collecte d'information sur l'élément intersecté, puis le calcul de la valeur de radiance du pixel en elle-même.

Cette étape de reconstruction va permettre de calculer la composante d'une image correspondant aux chemins lumineux LD^* (cf. Chap. 2-Sec. 1.4). Nous présenterons d'abord comment rassembler les informations, pour ensuite présenter comme les utiliser dans le calcul d'un valeur d'irradiance pour chaque pixel.

2.1 Initialisation d'un élément

Lorsqu'un élément de la hiérarchie de la solution de radiosité est passé pour la première fois à notre structure de reconstruction finale, toute l'information contenue sur ses liens propres ainsi que sur celle de ses parents est collectée. Tous les liens arrivant sur la surface parente originelle (celle qui n'a pas été subdivisé) sont rassemblés comme une information sur l'élément. Les autres liens servent à collecter les émetteurs en interactions avec cet élément et à initialiser les informations nécessaires à la reconstructions (nombre d'échantillons par sommet, facteur de forme, visibilité, etc), comme cela a été énoncé dans *[SSS01]*. Ces informations ne seront pas initialisés globalement

pour tous les sommets d'une scène, mais localement pour l'objet à considérer, permettant ainsi une approche ne travaillant que sur les éléments visibles dans l'image finale.

Sur les éléments

Les échanges qui ont lieux à haut niveau de la hiérarchie apportent en général une information d'éclairage "ambiant", alors que ceux qui sont bas dans la hiérarchie apportent plus d'informations de variations. Cela est vrai pour les critères de raffinement énergétiques (comme le *BFA* que nous employons), comme pour les critères plus évolués basés sur les variations ou des bornes d'énergie sur un élément (*[GH96, SSS97]*). C'est pourquoi, les liens représentant les échanges à haut niveau ne sont pas descendus pour calculer les variations en chacun des sommets. Ils sont accumulés sur la surface parente originelle de l'élément en considération.

Il est à noter que cela ne doit être fait qu'une seule fois par élément d'origine de la scène lorsque lui ou un de ses fils est rencontré par un rayon émanant du point de vue, et que cette approche réduit la quantité d'information par rapport à *[SSS01]*.

Sur les sommets

Sur les sommets v d'un élément de surface, tous les émetteurs e avec lesquels une interaction existe, vont être rassemblé, et pour chacun, une variation $\delta_{vis}(e,v)$ de la visibilité sur les éléments adjacents va être calculé, ainsi que celle du facteur de forme $\delta_{ff}(e,v)$. Ces émetteurs sont retrouvés à partir des liens arrivant sur cet élément ou ses parents, à l'exception de ceux déjà accumulés sur la surface parent originelle, comme expliqué précédemment.

Comme dans *[SSS01]*, le facteur de forme $ff_{e,v}$, entre l'émetteur e et le sommet v qui est en train d'être traité, est recalculé. En effet, ce calcul est bien moins lourd en général que celui de la visibilité, et peut donc être considéré comme négligeable. Cela permet d'avoir aussi une bonne représentation de sa variation à partir de l'interpolation des valeurs aux sommets. La variation de facteur de forme $\delta_{ff}(e,v)$ est alors :

$$\delta_{ff}(e,v) = Max_a(|ff_{e,v} - ff_{e,a}|) \tag{6.1}$$

où a représente un élément adjacent au sommet v, et $f_{e,a}$, le facteur de forme entre l'émetteur e et l'élément a.

La visibilité étant plus coûteuse à calculer, la moyenne des éléments adjacents est stocké sur le sommet v :

$$vis_{e,v} = \frac{1}{n_a} \sum_a vis_{e,a} \tag{6.2}$$

où n_a représente le nombre d'éléments adjacents au sommet, et $vis_{e,a}$ la visibilité entre l'élément adjacent a et l'émetteur e. La variation de visibilité devient alors :

$$\delta_{vis}(e,v) = Max_a(|vis_{e,v} - vis_{e,a}|) \tag{6.3}$$

Il est à noter que par rapport à *[SSS01]*, où le calcul de variation se faisait entre le sommet et ses sommets voisins, nous le faisons ici entre le sommet et les éléments voisins. Cela est essentiel pour notre approche restreinte aux éléments visibles dans une image. En effet, notre calcul va se limiter aux seuls sommets de l'élément visible, et n'en effectuer aucun sur les sommets des éléments voisins. Notre approche est complètement localisée.

Ces deux variations permettent de calculer deux erreurs, une sur la visibilité, soit $\Delta_{vis}(e,v)$, une autre sur le facteur de forme, soit $\Delta_{ff}(e,v)$:

$$\begin{cases} \Delta_{vis}(e,v) &= R_{max}\delta_{vis}(e,v)\,ff_{e,v}B_e \\ \Delta_{ff}(e,v) &= R_{max}vis_{e,v}\delta_{ff}(e,v)B_e \end{cases} \tag{6.4}$$

où R_{max} est la valeur maximale du coefficient de réflexion diffus sur la surface parente originelle considérée. Cela nous donne une borne maximale de l'erreur dans le cas ou le coefficient de réflexion n'est pas constant.

Ces deux erreurs permettent de déterminer si la variation de visibilité ou de facteur de forme est suffisamment importante pour nécessiter un nouveau calcul par échantillonnage pour chacun des points de l'élément traité. Le nombre d'échantillons nécessaires $nb_{v,e}$ est déterminé comme dans *[SSS01]*, par répartition du nombre total sur les émetteurs en fonction de l'irradiance qu'ils émettent, sans tenir compte de la visibilité.

2.2 Calcul de la valeur d'irradiance

Le calcul de la valeur d'irradiance pour un pixel comporte deux étapes principales, une évaluation à partir des liens et une évaluation à partir de l'information (qui sera définie dans la section ci-dessous) stockée sur les sommets. Cette valeur peut être ensuite réfléchie diffusément pour obtenir la valeur de radiosité qui sera par la suite utilisée dans le calcul de la valeur de radiance du pixel.

Avec les liens

Lorsque les informations concernant un élément ont été rassemblées, il devient possible de les utiliser pour calculer la valeur de l'éclairage en chacun des points qui sont visibles dans l'image finale. Lorsqu'il y a des informations sur les sommets disponibles, celles-ci sont utilisées prioritairement et l'irradiance due aux liens stockés est recalculée et ajoutée à celle calculée à partir des sommets. Dans le cas contraire, le facteur de forme et la visibilité (lorsqu'elle est déclaré partielle) sont réévalués au point d'intersection pour les liens les plus importants au point de vue contribution énergétique.

Solution de radiosité | Après la reconstruction finale

FIG. 6.2 : Reconstruction diffuse d'un bureau

Avec les sommets

L'ensemble des émetteurs en interaction avec l'élément est parcouru, et, pour chacun, une simple interpolation des valeurs de facteur de forme et de visibilité stockée sur chacun des sommets est faite lorsqu'il n'a été décidé qu'une réévaluation était nécessaire sur aucun des sommets. Dans le cas contraire, plusieurs possibilités se présentent.

Dans tous ces cas, une réévaluation de la visibilité et/ou du facteur de forme sera nécessaire. Le nombre d'échantillons à utiliser est déterminé par une interpolation du nombre décidé pour chacun des sommets comme dans *[SSS01]*. Sur un triangle dont les sommets sont numérotés $1,2,3$, le nombre d'échantillons nécessaires en un point p dont les coefficients d'interpolations sont $c_{1,2,3}$, est calculé suivant l'équation :

$$nb_{p,e} = c_1 \times nb_{v_1,e} + c_2 \times nb_{v_2,e} + c_3 \times nb_{v_3,e} \tag{6.5}$$

Les coefficients $c_{1,2,3}$ ont pour valeur l'aire du triangle opposé au sommet. Par exemple, c_1 est l'aire du triangle p, v_2, v_3. Lorsque $nb_{p,e}$ ne représente pas une valeur entière ($nb_{p,e} > \lfloor nb_{p,e} \rfloor$), une moyenne sur $\lfloor nb_{p,e} \rfloor + 1$ échantillons est effectuée, en affectant le poids $nb_{p,e} - \lfloor nb_{p,e} \rfloor$ au dernier échantillon. Cela permet de passer d'une valeur entière à une autre de façon continue.

Mais comme nous n'utilisons pas une variation calculée sur le sommets voisins, il se peut que pour un sommet, notre critère ait décidé qu'un nouvel échantillonnage était nécessaire, tandis que pour un sommet voisin, cela n'est pas nécessaire. À l'approche de ce dernier, le nombre d'échantillons donnés par l'interpolation devient donc nul.

Pour éviter cela, une moyenne est faite en utilisant les valeurs aux sommets pour ceux qui ne nécessitent aucun calcul supplémentaire, valeur pondérée par les coefficients d'interpolation respectifs, et la valeur recalculée par échantillonnage, pondérée par la somme des coefficients des autres sommets.

Par exemple, si pour le calcul de facteur de forme sur un triangle, le sommet 1 vérifie le critère de qualité, et pas les sommets 2 et 3, la valeur résultant de facteur de

forme devient

$$ff = c_1 \times ff_{v_1,e} + (c_2 + c_3) \times ff_{p,e} \tag{6.6}$$

où $(c_i, i = 1..3)$ représente les coefficients d'interpolation affectés à chacun des sommets, et $ff_{p,e}$, le facteur de forme recalculé par échantillonnage au point d'intersection p. Cela permet une transition continue entre les zones de recalcul et les zones d'interpolation seule.

3 Reconstruction et intégration des caustiques

L'algorithme unifié que nous avons présenté dans le chapitre Chap. 4 permet de traiter efficacement les chemins lumineux complexes, comme ceux générant des caustiques indirectes. Malgré l'utilisation de textures pour la représentation de ces effets (cf. Chap. 5), ce qui permet un affichage de bonne qualité pour un rendu interactif, cela n'est pas toujours suffisant pour obtenir la qualité attendue dans une image fixe. Nous allons présenter dans cette section des solutions pour intégrer l'information contenue dans les particules pour la représentation des caustiques, et plus généralement les effets $(L|D)S^+DE$ (cf. Chap. 2-Sec. 1.4), lors de notre reconstruction finale.

3.1 Utilisation des particules existantes

L'information des particules existantes peut être suffisante pour reconstruire les effets lumineux pixel par pixel. Notre approche pour la reconstruction va être similaire à celle du "Photon Map" *[Jen01]*, mais en restreignant les calculs aux seuls éléments visibles dans l'image. Ainsi la carte de photon ne va plus être globale, mais attaché à chaque élément surfacique originel de la scène. Cela permet de concentrer les calculs sur les seuls objets visibles.

Lorsqu'un élément est passé pour la première fois à notre structure de reconstruction finale, l'ensemble des impacts de particules appartenant à la surface parente originelle sont rassemblées, si cela n'a pas déjà été fait, et triées sous la forme d'un arbre binaire équilibré (cf. *[Jen01]*). Cela permet de trouver efficacement les particules au voisinage d'un point.

Ainsi pour chacun des points p de cet élément, sur lequel il faut évaluer l'énergie de la caustique, les N_p plus proches particules vont être recherchées avec les trois critères suivants :

$$\begin{cases} C_1 & = & R \leq R_{max} \\ C_2 & = & R \geq R_{min} \\ C_3 & = & N_p \geq N_{min} \end{cases} \tag{6.7}$$

Ici, R représente la plus grande distance à p pour les N_p particules, et N_{min}, le nombre minimal de particules souhaité dans le voisinage. C'est le rayon d'une sphère englobante des N_p particules, centrée en p. R est choisi pour que $N_p = N_{min}$, lorsque cela est possible. Les conditions C_1 et C_2 doivent par contre être toujours vrais. De plus, le

rayon minimal R_{min} peut être évaluer de façon à ce que la projection de la sphère englobante minimale couvre au moins un pixel de l'image final. Cela permettrait de réduire le bruit dans la reconstruction en utilisant une zone plus large que celle qui contiendrait seulement N_{min} particules, mais d'aussi d'effectuer le calcul sur l'ensemble des particules visibles dans un pixel.

L'estimateur utilisé, dérivé de celui présenté pour le "Photon Map" *[Jen96]*, avec une adaptation pour le calcul de l'irradiance, est alors de suivant :

$$I_c(p) = \frac{\pi R^2}{3} \sum_{i=1}^{N_p} \left(1 - \frac{r_i(p)}{R}\right) \phi_i \tag{6.8}$$

où $r_i(p)$ est la distance de la particule i au point p, et ϕ_i son énergie. Cette valeur est rajouté à l'irradiance calculée avec l'aide des liens, pour représenter l'ensemble des chemins $L(D|S^+)^*$ vers le point p.

3.2 Réémission de particules

Cependant, le nombre de particules présentes dans la scène peut être insuffisant pour la reconstruction de détails fins dans une image. Il faut donc pouvoir détecter les zones où une réémission est nécessaire afin d'accroître la qualité finale. Cette réémission va être, comme au chapitre précédent, guidé par les liens existants.

En effet, chacune des particules conserve l'information du lien à travers lequel elle a été émise. Ainsi, les N_p particules trouvé par la méthode précédente nous donne un ensemble de liens influençant par leurs particules la zone de reconstruction. Si ces particules ne sont pas suffisantes par leur nombre pour obtenir la qualité requise, un certain nombre est donc réémis à travers ces liens. Il faut cependant réajuster l'énergie pour l'adapter au nouveau nombre d'échantillons. Si, par exemple, un lien avait envoyé N_l particules et qu'il en renvoie N_l' nouvelles, l'énergie des anciennes particules est multiplié par

$$\frac{N_l}{N_l + N_l'} \tag{6.9}$$

et l'énergie des nouvelles particules est

$$\phi_{rs} = \frac{L(s,r) S_R S_S \mu(R)\ \mu(S) \tau R_r E_s}{\|s-r\|^2 (N_l + N_l')} \tag{6.10}$$

où s est un point choisi uniformément sur l'émetteur S d'un lien, s est un point choisi uniformément sur le récepteur R, μ est la mesure de l'objet (aire ou volume), R_r est 1 pour les groupes et $\cos\theta_i$ pour les surfaces, E_s est 1 pour les groupes et $\cos\theta_o$ pour les surfaces, F_S, F_R sont des facteurs d'échelles pour tenir compte des cas volumiques, valant 1 pour les surfaces et $\frac{1}{4}$ pour les groupes. Enfin, τ est l'absorption pour des milieux participatifs. Les valeurs $\cos\theta_i$ et $\cos\theta_o$ sont les cosinus de l'angle entre la direction de propagation et la normale respectivement de l'émetteur et du récepteur

(cf. Chap. 2-Sec. 2.4 et Chap. 2-Tab. 2.2). C'est une simple adaptation de l'équation Eq. 4.13 du chapitre Chap. 4.

Avec ces nouvelles particules, une nouvelle recherche est effectuées pour trouver les N_p plus proches comme décrit dans la section précédente. L'emploi du rayon minimal nous assure la convergence vers une solution. En effet, comme cela a été démontré par Walter *[Wal98]*, avec une reconstruction de ce type, la variance est à peu près égale à

$$Var(p) \simeq \frac{K(p)}{N_p R^2} \tag{6.11}$$

où $K(p)$ est une constante au point p. Ainsi, si le nombre de particules est doublé, le le rayon pour lequel les N_p plus proches particules sont disponibles est $R/\sqrt{2}$. Ainsi,

$$Var'(p) \simeq 2 * Var(p) \tag{6.12}$$

La variance est donc augmentée. Avec un rayon minimal, le nombre de particules utilisées va être augmenté sans pour autant réduire le rayon, et par là, la variance va être réduite.

Pour détecter si le nombre de particules est suffisant, nous allons employer le critère de contrôle de densité des particules introduit par Suykens *[SW00]*. Présenté dans le cadre du contrôle du nombre de particules à stocker, il définie un critère de convergence au point p par

$$C(p) = \frac{D_c(p)}{D_r} F\left(\frac{L_d + L_c}{L_c}\right) \tag{6.13}$$

où D_r est la densité à atteindre, $D_c(p)$ est la densité courante au point p, L_d l'énergie diffuse, L_c l'énergie de la caustique, et F est en général une puissance de 2 ou 3. Si $C(p) > 1$, le nombre de particules est considéré comme suffisant.

Ce critère favorise la densité du nombre de particules dans les zones où l'intensité lumineuse de la caustique est prédominante sur l'intensité lumineuse diffuse. Ainsi, dans les zones de faible influence des particules, peu ou pas de particules vont être renvoyées, tandis que dans les zones de forte influence, une réémission va se révéler plus souvent nécessaire afin d'accroître la qualité. Cela va permettre de concentrer l'effort de calcul dans les régions de l'image où cela va se révéler nécessaire, mais surtout, d'y accroître la qualité.

4 Composante dépendante du point de vue

Avec les deux étapes précédentes, il est possible de reconstruire l'ensemble de la composante indépendante du point de vue de l'image, c'est à dire les chemins $L(D|S)^*DE$. Il faut donc compléter la solution d'éclairage avec les effets dépendants du point de vue, c'est à dire les chemins $(L|D)S^+E$, comme les reflets spéculaires par exemple. Ceci sera fait par un lancer de rayons à partir du point de vue.

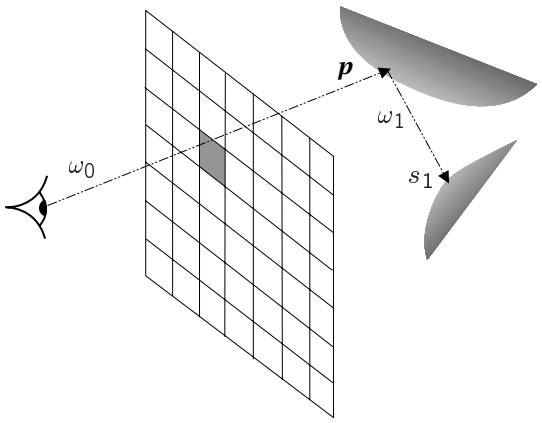


FIG. 6.3 : Émission d'un rayon à partir du point de vue

4.1 Utilisation d'un lancer de rayons

La quantité à estimer pour la composante dépendante du point de vue correspond donc aux chemins $(D|L)S^+E$. Cela correspond à l'équation

$$L(p,\omega_0) = \int_{\mathbb{S}} \rho_s(p,\omega_1,\omega_0) G'(p,s_1) L(s_1,\omega_1) ds_1 \tag{6.14}$$

où p est le point d'intersection d'un rayon émis à partir du point de vue dans la direction ω_0 (cf. Fig. 6.3), $\mathbb{S}$ est l'ensemble des points s_1 de la scène, ω_1 la direction de s_1 vers p, ρ_s la propriété de réflexion directionnelle, et $G'(p,s_l)$ un terme géométrique indiquant la proportion de l'énergie émise qui est reçue en p [1].

Or, la lumière émise par une surface $L(s,\omega_1)$, peut être décomposé par la somme des trois composantes, la composante diffuse $B(s_1)/\pi$ (cf. Chap. 2-Sec. 1.5), l'énergie propre émise de manière non-diffuse $L_{P_s}(s_1,\omega_1)$ et de l'énergie réfléchie par la composante directionnelle de la propriété matérielle $L_s(s_1,\omega_1)$:

$$L(s_1,\omega_1) = \frac{B(s_1)}{\pi} + L_{P_s}(s_1,\omega_1) + L_s(s_1,\omega_1) \tag{6.15}$$

La quantité $B(s_1)$ a déjà été calculée par notre méthode unifiée. La quantité $L_{P_s}(s_1,\omega_1)$ est elle une donnée initiale. Seule la valeur de $L_s(s_1,\omega_1)$ n'est pas connue. Elle correspond à l'intégrale

$$L_s(s_1,\omega_1) = \int_{\mathbb{S}} \rho_s(s_1,\omega_2,\omega_1) G'(s_1,s_2) L(s_2,\omega_2) ds_2 \tag{6.16}$$

Il est possible donc d'écrire à partir des équations Eq. 6.14, Eq. 6.15 et Eq. 6.16

$$L(p,\omega_0) = \int_{\mathbb{S}} \rho_s(p,\omega_1,\omega_0) G'(p,s_1) \left(\frac{B(s_1)}{\pi} + L_{P_s}(s_1,\omega_1) \right) ds_1 + P(p) \tag{6.17}$$

[1] Plus de détails sur les symboles de cette équation se trouvent dans le chapitre Chap. 2, sections Sec. 1.2, Sec. 1.5 et Sec. 3.1.

avec

$$P(p) = \int_{\mathbb{S}} \int_{\mathbb{S}} \rho_s(p, \omega_1, \omega_0) \rho_s(s_1, \omega_2, \omega_1) G'(s_1, s_2) L(s_2, \omega_2) ds_1 ds_2 \tag{6.18}$$

$P(p)$ représente des chemins directionnels de longueur supérieure à 2. Celui-ci peut être évalué en continuant la propagation du rayon. Une méthode de tracé de chemins s'arrêtant sur une réflexion diffuse (chemins $(L|D)S^+E$). Ainsi, la valeur de radiance calculée le long d'un rayon k de longueur $n >= 1$ est (cf. Chap. 2-Sec. 3.2)

$$E_k = \sum_{i=0}^{n-1} L_E(s_{i+1}, \omega_{i+1}) \Pi_{j=0}^{i} \frac{\rho_s(s_j, \omega_{j+1}, \omega_j) G'(s_j, s_{j+1})}{p(\omega_{j+1})} \tag{6.19}$$

où

$$L_E(s_{i+1}, \omega_{i+1}) = \frac{B(s_{i+1})}{\pi} + L_{P_s}(s_{i+1}, \omega_{i+1}) \tag{6.20}$$

et $p(\omega)$ est la densité de probabilité de réflexion dans la direction ω. Elle est en général choisie proportionnellement à $\rho_s(s_j, \omega, \omega_j)$. la longueur n est choisie telle que $n >= 1$, car il faut prendre en compte au moins une réflexion directionnelle, la réflexion diffuse ayant déjà été prise en compte dans les étapes précédentes.

4.2 Séparation direct/indirect

Pour simplifier la présentation des combinaisons dans les sections suivantes, nous avons réécris l'équation Eq. 6.19 pour un rayon k sous la forme :

$$\begin{aligned} E_k &= \frac{\rho_s(s_0, \omega_1, \omega_0) G'(s_0, s_1)}{p(\omega_1)} L_E(s_1, \omega_1) + \\ &\quad \frac{\rho_s(s_0, \omega_1, \omega_0) G'(s_0, s_1)}{p(\omega_1)} \sum_{i=1}^{n-1} L_E(s_{i+1}, \omega_{i+1}) \Pi_{j=1}^{i} \frac{\rho_s(s_j, \omega_{j+1}, \omega_j) G'(s_j, s_{j+1})}{p(\omega_{j+1})} \\ E_k &= E_{direct,k} + E_{indirect,k} \end{aligned} \tag{6.21}$$

$E_{direct,k}$ correspond à la réflexion directionnelle directe de l'environnement (chemins de longueur 1), tandis que $E_{indirect,k}$ correspond à des chemins de réflexions directionnelles d'une longueur supérieure.

4.3 Utilisation des directions des particules

Lorsque des particules sont présentes, elles contiennent une information sur les chemins $(L|D)S^+$. Dans la section Sec. 3, une réflexion diffuse nous a permis de calculer la valeur des caustiques. Mais une réflexion directionnelle permet d'obtenir aussi les chemins qui sont représentés par le terme $P(p)$. Il peut être directement calculé à partir des particules, comme cela est fait dans le "Photon Map" *[Jen96]* et par un estimateur similaire à Eq. 6.8 :

$$P(p) \simeq \frac{\pi R^2}{3} \sum_{i=1}^{N_p} \rho_s(p, d_i, \omega_0) \left(1 - \frac{r_i(p)}{R}\right) \phi_i \tag{6.22}$$

Dans cette équation d_i représente la direction incidente de la particule i. L'estimateur utilisé dans ce cas correspond ainsi à l'équation :

$$\begin{aligned} L(p,\omega_0) &\simeq \tfrac{1}{N_r}\textstyle\sum_{k=1}^{N_r} E_{direct,k} + \\ &\quad \tfrac{\pi R^2}{3}\textstyle\sum_{i=1}^{N_p} \rho_s(p,d_i,\omega_0)\left(1-\tfrac{r_i(p)}{R}\right)\phi_i \end{aligned} \tag{6.23}$$

Ainsi, grâce à une telle évaluation, la partie dépendante du point de vue est calculé avec l'aide d'une seule réflexions, ce qui peut être effectué dans des temps très réduits par rapport à la propagation complète de tous les rayons.

4.4 Combinaisons avec des rayons

Cependant, l'information fournie par les particules n'est pas toujours suffisante pour obtenir la qualité souhaitée. Nous allons ici proposer une méthode pour compléter cette information par celle obtenue par de nouveaux rayons.

Nous allons d'abord définir une erreur permettant de détecter quand un calcul est nécessaire. Nous proposons un critère basé sur la réduction de la variance. Lorsque la variance de l'estimation devient trop élevé, c'est à dire que

$$\left|\frac{\pi R^2}{3}\sum_{i=1}^{N_p}(\rho_s(p,d_i,\omega_0)\phi_i)^2\left(1-\frac{r_i(p)}{R}\right)-P(p)^2\right| > {\varepsilon_{max}}^2 \tag{6.24}$$

où ε_{max} est un paramètre donné par l'utilisateur, cette estimation peut être combinée *[SW99]* avec celle $P'(p)$ effectuée par la propagation de N_r rayons lancés à partir du point de vue, c'est à dire,

$$P'(p) = \frac{1}{N_r}\sum_{k=1}^{N_r} E_{indirect,k} \tag{6.25}$$

de la manière suivante :

$$\begin{aligned} L(p,\omega_0) &\simeq \tfrac{1}{N_r}\textstyle\sum_{k=1}^{N_r} E_{direct,k} + \\ &\quad \tfrac{N_p}{N_r+N_p}\tfrac{\pi R^2}{3}\textstyle\sum_{i=1}^{N_p}\rho_s(p,d_i,\omega_0)\left(1-\tfrac{r_i(p)}{R}\right)\phi_i + \\ &\quad \tfrac{1}{N_r+N_p}\textstyle\sum_{k=1}^{N_r} E_{indirect,k} \end{aligned} \tag{6.26}$$

La pondération effectué nous assure un bon comportement de la correction. Lorsque peu de rayons sont émis pour corriger la première estimation, par rapport aux nombres de particules, le calcul basé sur les particules est dominant. En effet, dans ce cas, comme le nombre de particules est supérieur à celui des rayons, elles apportent plus de détails que les rayons. Dans le cas où le nombre de rayons est supérieur à celui des particules, leur influence devient ainsi prépondérante. Les particules sont ici considérées comme des rayons déjà propagés. Les particules et les rayons combinés améliore la qualité finale.

Solution de radiosité lissée
12 secondes

Après la reconstruction finale
300 secondes

195 polygones initiaux, 6 500 polygones après subdivision, image 640x480

FIG. 6.4 : Reconstruction diffuse d'un salon

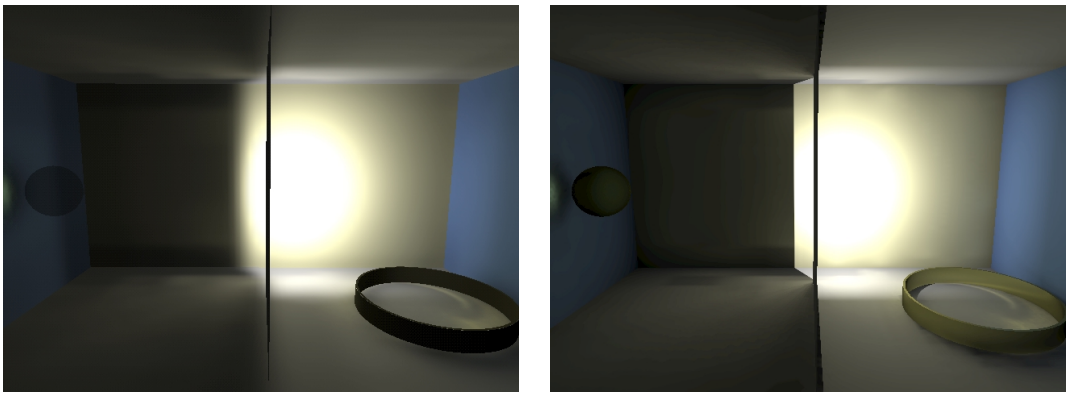
5 Premiers résultats

Nous avons implémenté les méthodes décrites dans les sections Sec. 2, Sec. 3.1 et Sec. 4.1 pour expérimenter des premiers résultats. Tous les tests ont été effectués sur une station SGI-Octane2 (processeur MIPS R12000 à 400Mhz). Afin d'accroître la qualité des résultats, nous avons implémenté un méthode d'anti-crénellage simple. Un certain nombre de rayons choisi par l'utilisateur est envoyé dans l'environnement. Si les surfaces intersectées sont identiques, une seule valeur est calculée. Sinon, une moyenne des valeurs de chacun des rayons est effectuée.

Ces résultats montre le potentiel dans le calcul d'images de très grande qualité, notamment due aux fait que la composante diffuse d'une scène n'est pas bruitée. De plus, cette approche permet de traiter efficacement les cas indirect, comme cela va être démontré dans les tests présentés ici. Il est à noter qu'aucune de nos structures de calculs n'a été optimisé, notamment le tracé de rayons qui reste basé sur l'emploi de la hiérarchie de boîte englobante. L'utilisation d'une grille, voir une grille récursive pourrait donner des biens meilleurs résultats en ce qui concernent la rapidité de calcul.

5.1 Diffus seul

Nous avons effectuées nos premiers tests sur des environnements diffus. Un premier résultat est visible dans la figure Fig. 6.2. La différence de qualité est nettement visible, notamment au point de vue des ombres qui sont bien représentées. Un autre exemple avec des temps de calcul est visible dans la figure Fig. 6.4. Il est ainsi possible d'obtenir des images de grande qualité dans un temps de calcul raisonnable. De plus, de par la structure dépendante du point de vue, il est possible de traiter des scènes



Méthode unifiée (lissage+textures) — Reconstruction finale

24 secondes — 13 minutes

13 objets initiaux, 11 105 objets après subdivision, image 640x480

FIG. 6.5 : Scène avec deux pièces en éclairage indirecte.

complexes avec des ressources mémoires limitées (200 Mo pour Fig. 6.9). La solution s'adapte bien à une éclairage diffus majoritairement indirect (cf. Fig. 6.8).

5.2 Caustiques et Composante dépendante du point de vue

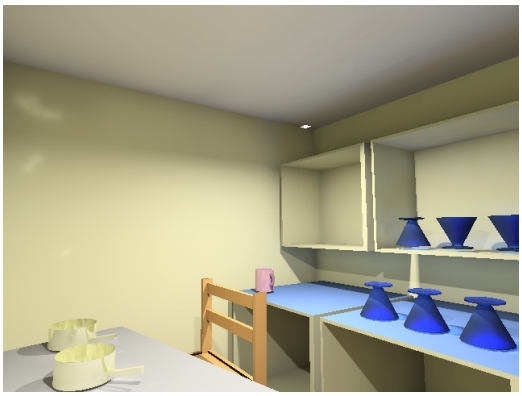
Nous avons testé la méthode présentée dans la section Sec. 3.1 et celle présentée dans la section Sec. 4.1 sur une scène simple avec un éclairage principalement indirect (cf. Fig. 6.5), où les chemins lumineux sont difficiles à détecter par des méthodes classiques. L'ensemble de l'environnement est éclairé par un source orientée vers le mur du fond, d'où provient l'éclairage principale de la scène. Une caustique est visible sur le sol de la pièce de droite due à l'anneau, et sur le mur de gauche, à cause de la sphère.

La seconde scène est une salle à manger (cf. Fig. 6.6) contenant plusieurs objets créant des caustiques, comme des verres et des casseroles métallisées. Cette scène permet de tester la combinaison de toutes les méthodes de reconstruction.

Afin de tester la combinaison des particules et des rayons pour la composante spéculaire indirecte, nous avons utilisé un scène d'un couloir comportant un sol ayant une composante directionnelle moyenne, et sur les mur, des tableaux blancs. Il est ainsi possible de distinguer aisément des reflets sur le sol correspondant à la réflexion des plafonniers sur les tableaux blancs. Dans la figure 6.10, nous montrons trois types de reconstruction pour la composante dépendante du point de vue : une basée sur la seule utilisation des rayons, la suivante sur l'utilisation des seules particules pour le directionnel indirect, et la dernière avec une combinaison des particules et des rayons.

Nous avons effectué un premier test de notre critère de renvoi de particules. Pour cela, nous avons utilisé une scène simple, composée d'un anneau reflectif dans un environnement purement diffus. La source de lumière crée une caustique bien connue,

Méthode unifiée (lissage+textures)
17 secondes



Reconstruction finale
37 minutes

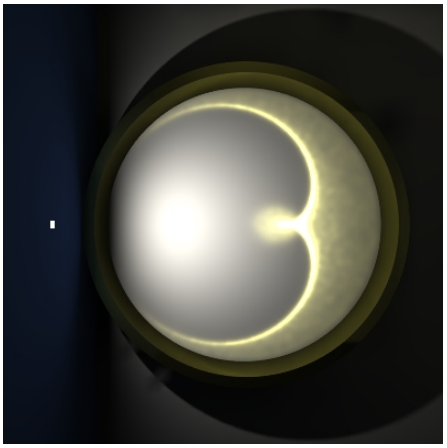
483 objets initiaux, 30 000 objets après subdivision, image 640x480

FIG. 6.6 : Salle à manger en éclairage directe.

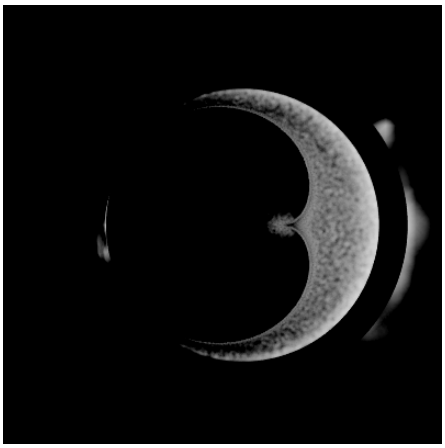
ayant la forme d'une cardioïde. Nous présentons dans la figure Fig. 6.7 à la fois l'image générée, mais aussi une image représentant les zones de réémission nécessaire.

5.3 Scènes complexes

Nous avons appliqué notre approche dans le calcul d'images sur des scènes complexes, afin d'éprouver notre approche dépendante du point de vue. Ces tests montrent que de telles approches permettent une solution en limitant la mémoire requise. La première scène (cf. Fig. 6.8) est la scène "Soda Shoppe" modifiée avec des sources



Reconstruction finale



Zones de renvoi

FIG. 6.7 : Test du critère de réémission.

Vue générale - Solution de radiosité
170 secondes

Détail d'un table - Reconstruction finale
38 minutes

1781 objets initiaux, 44 670 objets après subdivision, image 640x480

FIG. 6.8 : Reconstruction diffuse du "Soda Shoppe" dans un éclairage indirecte.

de lumières orientées vers le plafond pour forcer une éclairage indirect sur une scène complexe diffuse.

La deuxième scène est un intérieur de bateau (cf. Fig. 6.9). Cette scène est composé principalement de petits objets, notamment pour la description des tuyaux et des vannes. Ce test montre que l'algorithme réagit bien à une telle complexité.

Un couloir
2 heures et 32 minutes

Ensemble de vannes
2 heures et 27 minutes

167 708 objets initiaux, 182 933 objets subdivisés
23 minutes pour la solution globale

FIG. 6.9 : Deux points de vue de l'intérieur d'un bateau

6 Conclusion

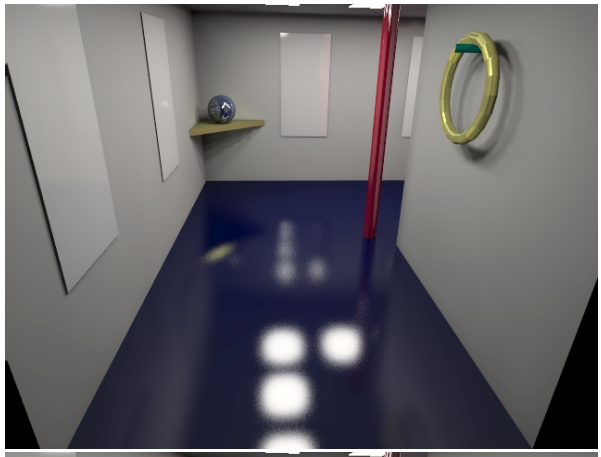
Dans ce chapitre, nous avons introduit une nouvelle approche locale pour la reconstruction finale pour notre algorithme unifié. L'algorithme unifié et les méthodes de reconstruction des chapitres précédents nous fournissent des informations nécessaires à l'obtention d'une image de très grande qualité. Il peut être considéré comme un précalcul guidant le rendu d'une image.

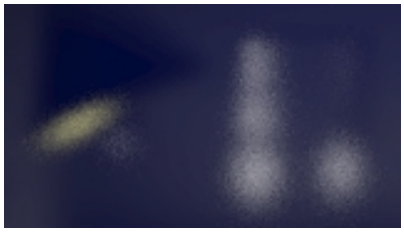
Nous avons présenté une nouvelle reconstruction diffuse basée sur les travaux de Scheel et al. *[SSS01]*, que nous avons adapté à notre approche locale. Notre solution est moins consommatrice de ressources mémoire et permettrait de la gérer efficacement par une méthode de cache. L'architecture proposée nous permet de tirer profit de l'approche dépendante du point de vue et d'ainsi restreindre la quantité d'information qu'il faut récupérer pour calculer une solution. Nous avons aussi introduit un méthode permettant la reconstruction des caustiques à partir des particules présentes et une solution pour améliorer la qualité de ces effets lumineux par le renvoi de particules lorsque cela est nécessaire. Pour finir, nous avons présenté des méthodes permettant l'intégration de la composante dépendante du point de vue, par simple lancer de rayon, par l'utilisation des particules existantes, et aussi par l'intégration de ces deux types de calculs.

Cette approche pixel par pixel permettrait aussi une parallélisation sur tout type d'architecture, et notamment celles à mémoire distribuée de type "grappe". En effet, il est possible de segmenter l'image et de répartir le calcul sur plusieurs ordinateurs. La réparation des informations nécessaires se fera automatiquement.

Les résultats présentées ici démontre qu'une telle approche possède un grand potentiel dans l'obtention d'images de très grande qualité, des images peu bruitées, contenant toutes les composantes de la solution d'éclairage global. Ils soutiennent le développement de cette méthode.

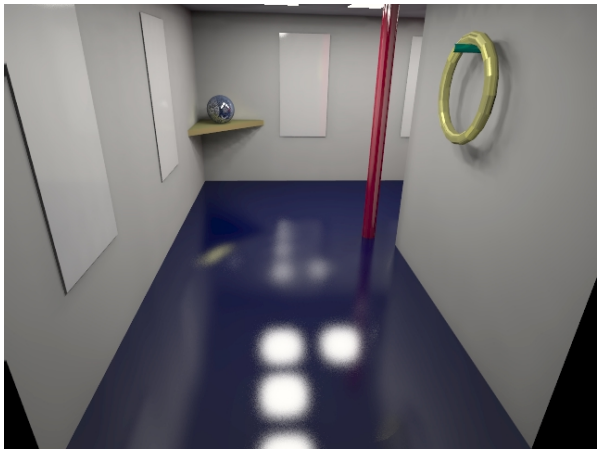
Cependant, nous avons ici introduit de nouveaux paramètres. Le contrôle de la qualité est possible, mais complexe. Les temps de calcul permettent souvent de relancer le rendu d'un solution avec de nouveaux paramètres, mais il faudrait trouver le moyen de simplifier tout cela. Il a tout de même une garantie que lorsque ces paramètres sont à leur maximum, la qualité sera optimale.

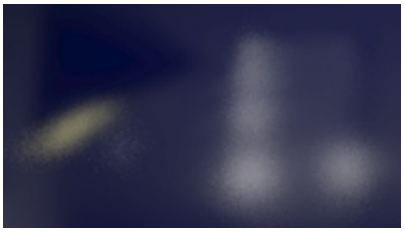




Rayons
4 heures

Particules
3 heures et 53 minutes





Rayons + Particules
4 heures et 10 minutes

424 objets initiaux, 7 454 objets après subdivision, image 640x480
500 particules pour la reconstruction, 800 rayons par pixel , 9 échantillons d'anti-crénellage

FIG. 6.10 : Scène de couloir.

7

Conclusion

Au fil de ce document, nous avons présenté une nouvelle approche pour le contrôle automatique de la qualité d'un calcul d'éclairage global. Nous allons ici récapituler ces différentes étapes. Nous allons aussi en profiter pour donner quelques pistes issues de notre réflexion à la suite du développement de notre approche.

Résumé des contributions

Dans ce document, nous avons démontré que, par une caractérisation des différents types d'échanges lumineux et par leur décomposition, il est possible de fournir des algorithmes unifiés, permettant de combiner différentes méthodes existantes afin de les utiliser dans leur(s) domaine(s) de prédilection.

Le nouvel algorithme que nous avons proposé est une approche permettant une intégration d'une méthode de radiosité hiérarchique avec regroupement et d'un lancer de particules. Le lien, structure créée par l'algorithme hiérarchique et représentant une interaction entre deux éléments hiérarchiques, permet la détection automatique des différentes catégories d'échanges. Ainsi, pour tout échange diffus, la radiosité reste utilisée, mais lorsqu'une composante non diffuse existe, un lancer de particules est effectué, ceci de façon complètement intégrée dans les étapes d'un algorithme de radiosité hiérarchique avec regroupement.

De part la nature même d'un algorithme hiérarchique, cette méthode permet une transition d'une solution rapide et préliminaire vers une solution de plus grande qualité. Elle est de plus efficace dans le traitement de chemins lumineux compliqués, permettant par exemple le calcul d'effets lumineux majoritairement indirects. Cette efficacité nous permet d'un coté, de traiter des scènes complexes et d'obtenir des solutions dont le bruit est très réduit par rapport aux méthodes probabilistes pures, et, d'un autre coté, une certaine interactivité pour des scènes simples.

Pour obtenir des caustiques de meilleure qualité, nous avons proposée une méthode utilisant une reconstruction mixte, combinant l'emploi d'un maillage pour la représentation des effets lumineux de faible fréquence spatiale, et celui de textures pour les effets avec de fortes variations.

Afin d'étendre l'algorithme vers un cadre dynamique, nous avons proposé un méthode permettant de détecter quelles sont les particules à réémettre lorsqu'un objet bouge. Cette méthode est basée sur un octree dynamique localisant les régions de l'espace affectées par le mouvement, et sur la structure des liens pour permettre une détection des échanges diffus, mais aussi pour détecter quelles ont les particules à réémettre. Elle permet une mise à jour rapide, voire interactive mais avec dégradation de qualité.

Pour obtenir une solution de très grande qualité, dans le cadre du calcul d'une image, nous avons dans le dernier chapitre présenté une première approche au problème de la reconstruction finale. La méthode, principalement dépendante du point de vue, tire partie des informations existantes dans une solution de notre méthode unifiée, afin de reconstruire le plus précisément possible l'ensemble des effets. Ces approches combinent divers types de reconstructions, basé sur les liens, sur les particules et sur lancer de rayons afin d'accroître la qualité de la solution existante, lorsque cela se trouve être nécessaire.

Une des lignes directrices des toutes ces solutions proposées, c'est la décomposition des chemins lumineux et l'intégrations des méthodes. Chaque méthode possède un ensemble de chemins lumineux pour lesquelles elles excellent. La décomposition permet donc de l'employer dans son domaine de prédilection. Le second point est donc l'intégration. Chaque méthode fournit une certaine quantité d'information dont les autres méthodes doivent pouvoir bénéficier. Les étapes importantes de la mise en place d'une telle méthode sont la séparation minutieuse des chemins lumineux, puis le choix des méthodes et la définition précise de tout ce qu'elles peuvent transmettre les unes aux autres, tout en conservant une méthode unifiée et générale.

Le développement de ces méthodes a été possible grâce à la création d'une plateforme d'expérimentation. Celle-ci propose une architecture généraliste permettant une mise en place facilité d'algorithmes d'éclairage divers, de même que leur unification au sein d'une même méthode.

Travaux futurs

Notre approche de décomposition des chemins lumineux nous a permit de séparer la composante indépendante du point de vue, que notre méthode unifiée calcule efficacement, de la composante dépendante du point de vue. La première peut être rendu interactivement en utilisant le matériel graphique, et nous avons présenté une méthode permettant son rendu interactif. Il faut cependant améliorer la qualité comme la vitesse de rendu de cette composante, en précalculant certains données ou/et en utilisant plus le matérielle graphique disponible.

La méthode que nous avons présentée constitue une première étude d'une intégration des algorithmes. Malgré tout, nous n'avons pas utilisé à leur plein potentiel chacune des méthodes employées, que cela soit la radiosité hiérarchique avec regroupement, le lancer de photon, ou le lancer de rayons. Une meilleure intégration pourrait être ainsi obtenue. De plus, cette approche a montré sont efficacité, d'autres combinaisons pourrait se montrer plus efficaces.

Notre méthode ne prend pas en compte les milieux participatifs comme le brouillard, les fumées, et c'est une de ses limitations actuelles. Il pourrait cependant être étendue pour ce domaine, par une approche similaire de séparation des chemins lumineux. Un milieu participatif peut être considéré comme absorbant et diffusant.

La méthode que nous avons présenté, mise à part le cas dynamique et la reconstruction finale, possède un petit nombre de paramètres, chacun avec une influence bien spécifique. Ce nombre de paramètres explose vite lorsqu'elle est adapté à certains cas particuliers. Le contrôle s'en retrouve compliqué. Il serait intéressant de trouver un moyen d'automatiser le choix des paramètres, mais la tâche est extrêmement difficile. Un moyen pour faciliter le contrôle serait d'utiliser l'expérience d'un utilisateur dans sa perception et sa connaissance d'une solution d'éclairage. Les indications qu'il pourrait fournir sur la solution finale pourrait guider et faciliter le calcul.

Une des notions qui a permis de mettre en place cet approche, est la notion de flux. Elle permet de bien appréhender les échanges lumineux dans leur phénomène de diffuse. Cette approche de flux a déjà été abordé *[LF96]*, et il serait intéressant de plus la développer. Elle permettrait notamment de calculer une solution complètement indépendante de la géométrie de la scène.

Pour conclure

La mise en place de nos approches a été l'occasion d'une profonde réflexion sur l'état de l'éclairage globale à l'heure actuel et sur ses enjeux. La plupart des solutions sont là, mais trop indépendantes les unes des autres pour être exploitées facilement. Nos travaux font un premier pas vers leur intégration. La facilité d'accès reste un grand enjeux de l'ensemble de ces méthodes. L'éclairage global est souvent associé à qualité, mais trop souvent aussi à complexité. Un effort de simplification dans l'algorithme et les structures de données, et un effort d'intégration sont donc essentiels à ce jour pour fournir une méthode d'approche plus intuitive.

Annexes

A

Paramétrisation des objets

Nous allons détailler ici brièvement les paramétrisations choisies pour les objets comme des cônes, cylindre, sphère, anneau. Bien que chaque type d'objet pose des problèmes spécifiques, il y a une manière globale de les aborder. Comme il a été dit dans le chapitre 3. Il est par exemple important d'assurer la continuité et l'unicité de la paramétrisation, afin de faciliter les partages de sommets et la subdivision des éléments de discrétisation.

1 Sphère et portion de sphère

Choix de paramétrisation

La première paramétrisation d'une sphère qui vient à l'esprit est basée sur les coordonnées sphériques. Cependant, une telle représentation souffre d'un défaut majeur, les discontinuités qui apparaissent aux pôles.

La paramétrisation ici choisie utilise les vecteurs de directions. Un point sur la sphère est alors $p = O + R\vec{d}$, et un triangle sphérique est déterminée par trois direction. Chaque point sur la sphère est ainsi paramétrisé de manière unique par sa direction $\vec{d} = \frac{p-O}{R}$.

Pour avoir une subdivision hiérarchique régulière, il faudrait que la première subdivision soit un isododécaèdre, ce qui du point de vue hiérarchique es très mauvais car il y a création de trop d'éléments en une seule subdivision. Nous utiliserons donc pour la première subdivision un octaèdre. La subdivision des triangles sphériques ainsi obtenus se fait comme sur des triangles classiques.

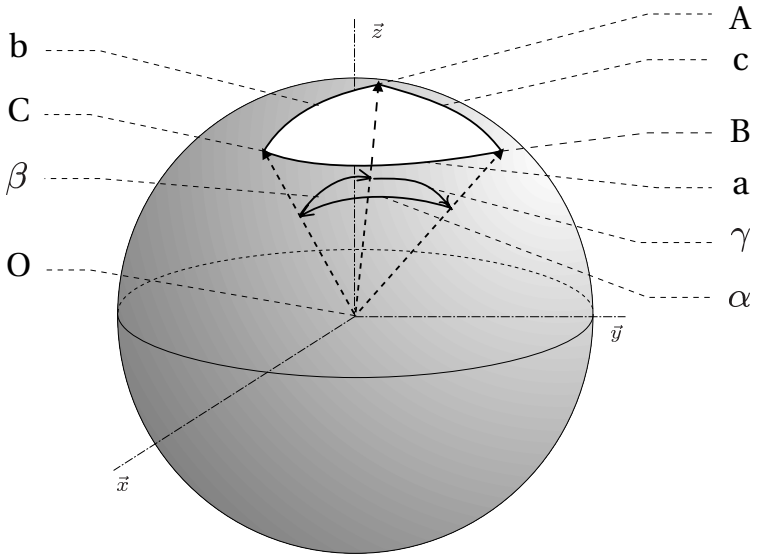


FIG. A.1 : Représentation d'une portion de sphère

Calcul de l'aire des portions de sphères

Cela nous est donnée d'après le théorème de Girard :

$$A = R^2(\alpha+\beta+\gamma-\pi) \tag{A.1}$$

avec

$$\begin{cases} \cos(\alpha) &= <[B|A] \cdot [C|A]> \\ \cos(\beta) &= <[A|B] \cdot [C|B]> \\ \cos(\gamma) &= <[B|C] \cdot [A|C]> \end{cases} \tag{A.2}$$

où l'opérateur $[.|.]$ représente

$$[u|v] \;=\; Normaliser(u - <u \cdot v> v) \tag{A.3}$$

Échantillonnage uniforme

Sur la sphère, le problème est assez simple. Étant donnée deux nombres aléatoires, soit $\varepsilon_1 \in [0,1[$ et $\varepsilon_2 \in [0,1[$, un point p sur la sphère suivant une loi uniforme peut être calculé par :

$$p \;=\; R \begin{bmatrix} 2\cos(2\pi\varepsilon_2)\sqrt{\varepsilon_1(1-\varepsilon_1)} \\ 2\sin(2\pi\varepsilon_2)\sqrt{\varepsilon_1(1-\varepsilon_1)} \\ 1-2\varepsilon_1 \end{bmatrix} \tag{A.4}$$

Sur un triangle sphérique, le problème se retrouve bien plus complexe. Arvo en propose une solution dans *[Arv95]*. Soit, en gardant les notations de la figure Fig. A.1 :

$$
\begin{aligned}
\hat{A} &= \varepsilon_1 A \\
s &= \sin(\hat{A}-\alpha) \\
t &= \cos(\hat{A}-\alpha) \\
u &= t-\cos(\alpha) \\
v &= s+\sin(\alpha)\cos(c) \\
q &= \frac{(v\,t-u\,s)\cos(\alpha)-v}{(v\,s+u\,t)sin(\alpha)} \\
\hat{C} &= q\,A+[C|A]\sqrt{1-q^2} \\
z &= 1-\varepsilon_2(1-<\hat{C}\cdot B>) \\
p &= R(zB+[\hat{C}|B]\sqrt{1-z^2})
\end{aligned}
\tag{A.5}
$$

2 Cône et portion de cône

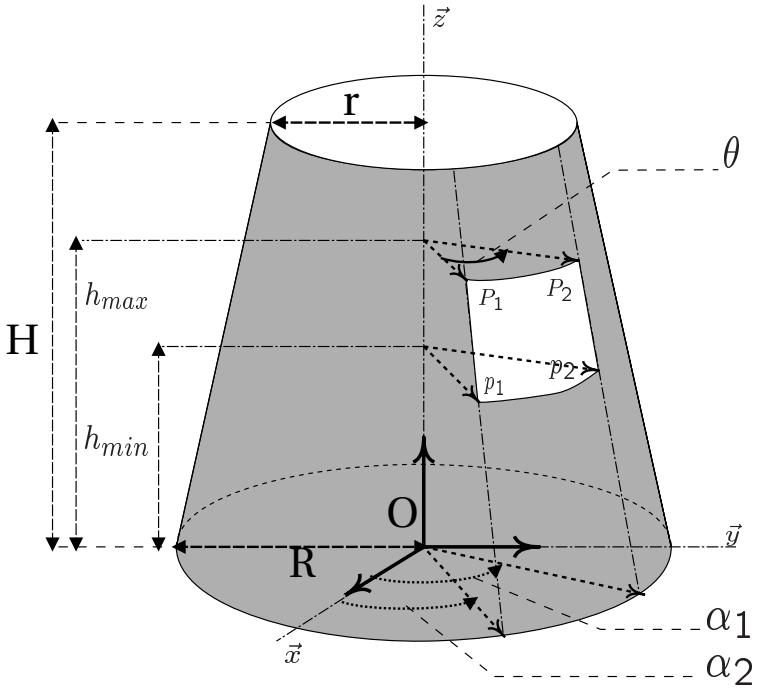


FIG. A.2 : Représentation d'une portion de cône

Choix de paramétrisation

La paramétrisation implicite d'un cône réside dans les deux paramètres, hauteur sur l'axe $\vec{z}$ et angle par rapport à $\vec{x}$, ou un rayon et l'angle (cf. Fig. A.2). Cependant, comme pour la sphère une telle représentation souffre d'un défaut majeur, la discontinuité qui apparaît sur le paramètre d'angle.

La paramétrisation ici choisie utilise aussi les vecteurs de directions dans le plan de section orthogonal à la direction principale $\vec{z}$ (cf. Fig. A.2). Un point sur le cône est alors représenté par une direction $\vec{d} = (d_x, d_y)$ et sa hauteur h sur l'axe $\vec{z}$

$$p = O + h\vec{z} + \left(\frac{h(r-R)}{H} + R\right)(d_x\vec{x} + d_y\vec{y}) \tag{A.6}$$

et une portion de cône est définie par quatre de ces positions P_1,P_2, p_1 et p_2. Mise à part le cas particulier où $r = 0$, chaque point p du cône est ainsi définie de manière unique.

$$\begin{array}{rcl} h & = & < (p-O) \cdot \vec{z} > \\ d_x & = & \frac{H}{h(r-R)+H\times R} < (p-O) \cdot \vec{x} > \\ d_y & = & \frac{H}{h(r-R)+H\times R} < (p-O) \cdot \vec{y} > \end{array} \tag{A.7}$$

La normale à la surface en ce point p est alors :

$$\vec{n} = \frac{H(d_x\vec{x} + d_y\vec{y}) + (R-r)\vec{z}}{\sqrt{H^2 + (R-r)^2}} \tag{A.8}$$

Par la suite, l'hypothèse émise est que pour une portion de cône, P_1 et p_1 sont sur une même génératrice (de même que P_2 et p_2), c'est à dire qu'ils ont la même direction $\vec{d_1}$ (respectivement $\vec{d_2}$), et que p_1 et p_2 sont sur une même plan orthogonal à la direction principale $\vec{z}$ (de même que P_1 et P_2), c'est à dire qu'ils sont à la même hauteur h_{min} (respectivement h_{max}).

Calcul des aires

L'aire d'un cône tronqué est défini par l'équation

$$A = \pi(R^2 - r^2)\sqrt{\left(\frac{H}{R-r}\right)^2 + 1} \tag{A.9}$$

et avec les hypothèses formulées ci-dessus, l'aire d'une portion de cône est

$$A = \theta(r_{max}^2 - r_{min}^2)\sqrt{\left(\frac{H}{R-r}\right)^2 + 1} \tag{A.10}$$

avec

$$\begin{aligned} \theta &= \cos^{-1}(< \vec{d_1} \cdot \vec{d_2} > \\ r_{min} &= R + \frac{h_{max}(r-R)}{H} \\ r_{max} &= R + \frac{h_{min}(r-R)}{H} \end{aligned} \tag{A.11}$$

Échantillonnage uniforme

Si ε_1 et ε_2 sont deux valeurs tirées selon une loi uniforme dans l'intervalle $[0, 1]$, alors il est possible de définir une position p selon une loi uniforme sur un cône

$$\begin{aligned} \phi &= 2\varepsilon_1\pi \\ a &= \sqrt{\varepsilon_2(R^2 - r^2) + r^2} \\ h &= \frac{H(R-a)}{R-r} \\ p &= 0 + h\vec{z} + a(\cos(\phi)\vec{x} + \sin(\phi)\vec{y}) \end{aligned} \tag{A.12}$$

ou sur une portion de cône comme cela a été défini ci-dessus

$$\begin{aligned} \phi &= \varepsilon_1 \theta \\ X &= \cos(\phi) d_{1x} - \sin(\phi) d_{1y} \\ Y &= \cos(\phi) d_{1y} + \sin(\phi) d_{1x} \\ d &= \sqrt{(r_{max}^2 - r_{min}^2)\varepsilon_2 + r_m in^2} \\ h &= \frac{(R-d)H}{R-r} \\ p &= o + h\vec{z} + d(X\vec{x} + Y\vec{y}) \end{aligned} \tag{A.13}$$

Subdivision uniforme

Le cône est toujours subdivisé en quatre quadrants selon l'angle, pour retrouver une information de directionnalité pour les échange, indisponible sur le cône complet.

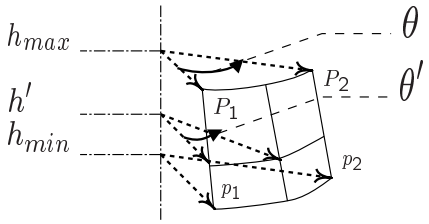


FIG. A.3 : Représentation d'une portion de cône

La subdivision d'une portion de cône se fait comme pour un "quadtree". Il est subdivisé en quatre fils d'aires équivalentes. Pour cela, la portion est coupée en deux pour le paramètre angulaire $\theta' = 0.5 * \theta$, la hauteur au centre h' est choisie comme suit (cf. Fig. A.3) :

$$\begin{aligned} d &= \sqrt{\frac{r_{max}^2 + r_{rmin}^2}{2}} \\ h' &= \frac{(R-d)H}{R-r} \end{aligned} \tag{A.14}$$

3 Cylindre et portion de cylindre

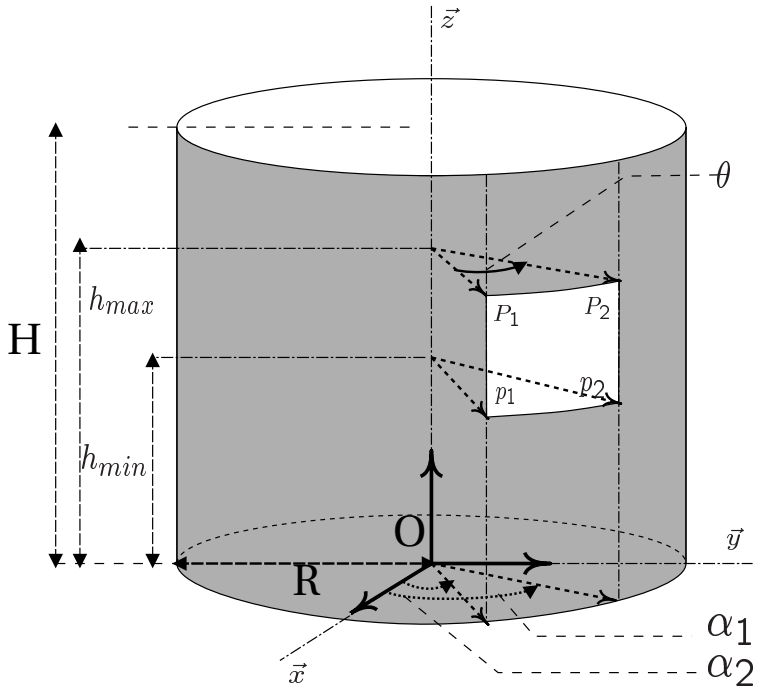


FIG. A.4 : Représentation d'une portion de cylindre

Choix de paramétrisation

La paramétrisation du cylindre est très proche de celle du cône. Chaque point p sur la surface est représenté par une direction à l'axe du cylindre $\vec{d}$ et la hauteur sur l'axe h :

$$p = O + h\vec{z} + R(d_x\vec{x} + d_y\vec{y}) \tag{A.15}$$

Chaque point sur le cylindre est ainsi défini de manière unique

$$\begin{array}{rcl} h & = & < (p-O) \cdot \vec{z} > \\ d_x & = & \frac{1}{R} < (p-O) \cdot \vec{x} > \\ d_y & = & \frac{1}{R} < (p-O) \cdot \vec{y} > \end{array} \tag{A.16}$$

et la normale en ce point est

$$\vec{n} = d_x\vec{x} + d_y\vec{y} \tag{A.17}$$

De même, une portion de cylindre est définie par quatre points P_1,P_2, p_1 et p_2. Pour la simplicité de calcul, les mêmes hypothèses que pour le cône sont faites.

Calcul de l'aire d'une portion de cylindre

Avec les hypothèses faites, l'aire d'une portion de cylindre est

$$A = \theta R(h_{max} - h_{min}) \tag{A.18}$$

avec

$$\theta = \cos^{-1}(< \vec{d_1} \cdot \vec{d_2} > \tag{A.19}$$

Échantillonnage uniforme

Avec ces hypothèses, un échantillonage uniforme pour le cylindre comme pour une de ces portions se fait par un échantillonnage uniforme de l'angle et de la hauteur.

Subdivision uniforme

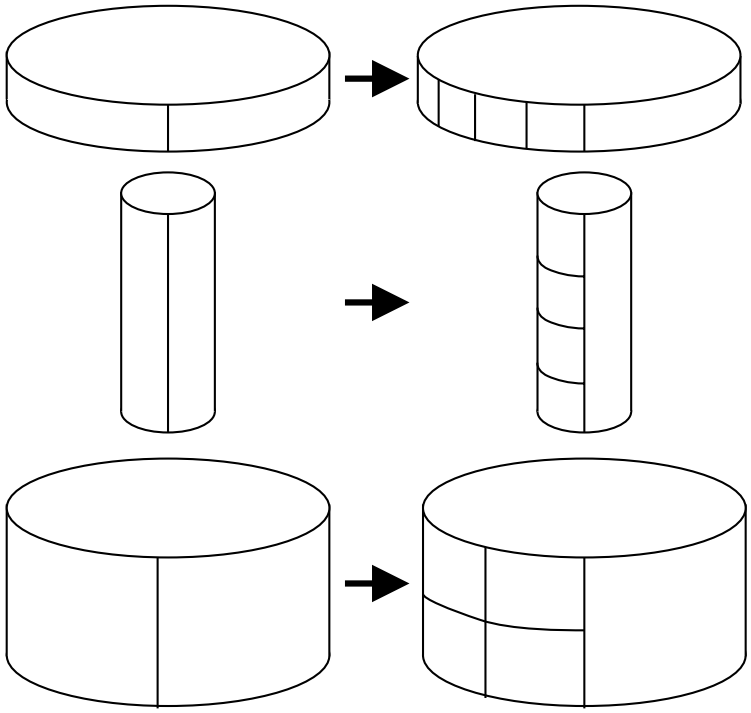


FIG. A.5 : Subdivision d'une portion de cylindre

Comme pour le cône, le cylindre est toujours subdivisé en quatre quadrants suivant l'angle (et jamais dans le sens de la hauteur).

Pour les portions de cylindres, il faut essayer tant que possible d'éviter les maillages trop allongés. Pour cela, deux tests sont définis pour trouver ces cas. Le premier cherche à vérifier si la hauteur n'est pas trop grande.

$$_3\theta R < (h_{max} - h_{min}) \tag{A.20}$$

Si il est vérifié, la subdivision se fera dans le sens de la hauteur, en quatre éléments (cf. Fig. A.5 au centre). Le second test fait la même vérification, mais selon la dimension angulaire.

$$_3(h_{max} - h_{min}) < \theta R \tag{A.21}$$

Si il est vérifié, la subdivisions se fera dans le sens angulaire, en quatre éléments (cf. Fig. A.5 en haut). Dans le cas ou aucun des deux tests n'est validé, la subdivision se fait comme un quadtree (cf. Fig. A.5 en bas).

4 Anneau et portion d'anneau

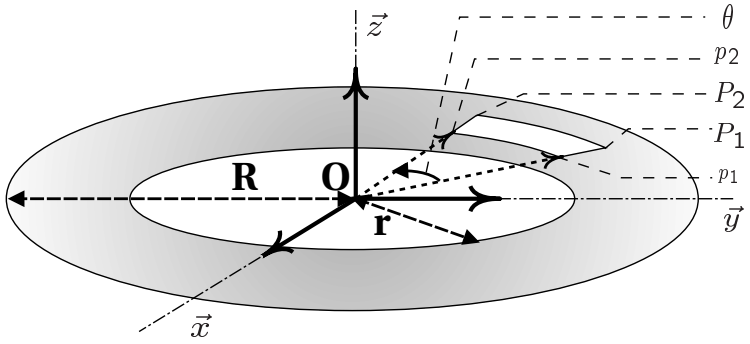


FIG. A.6 : Représentation d'une portion d'anneau

Choix de paramétrisation

La paramétrisation ici choisie est très proche de celle du cône. Un point sur l'anneau est ici représenté par une direction $\vec{d} = (d_x, d_y)$ et son rayon r au centre O (il n'y a pas de hauteur sur l'anneau)

$$p = O + h\vec{z} + \left(\frac{h(r-R)}{H} + R\right)(d_x\vec{x} + d_y\vec{y}) \tag{A.22}$$

et une portion d'anneau est définie par quatre de ces positions P_1,P_2, p_1 et p_2. Mise à part le cas particulier où $r = 0$, chaque point p du l'anneau est ainsi défini de manière unique, sauf au centre :

$$\begin{array}{rcl} a & = & < (p-O) \cdot \vec{x} > \\ b & = & < (p-O) \cdot \vec{y} > \\ r & = & \sqrt{a^2+b^2} \\ d_x & = & a/R \\ d_y & = & b/R \end{array} \tag{A.23}$$

La normale à la surface en ce point p est alors constante :

$$\vec{n} = \vec{z} \tag{A.24}$$

Par la suite, l'hypothèse émise est que pour une portion d'anneau, P_1 et p_1 sont sur une même génératrice (de même que P_2 et p_2), c'est à dire qu'ils ont la même direction $\vec{d_1}$ (respectivement $\vec{d_2}$), et que p_1 et p_2 sont sur une même cercle (de même que P_1 et P_2), c'est à dire qu'il ont le même rayon r_{min} (respectivement r_{max}).

Calcul des aires

L'aire d'un anneau tronqué est défini par l'équation

$$A = \pi(R^2 - r^2) \tag{A.25}$$

et avec les hypothèses formulées ci-dessus, l'aire d'une portion d'anneau est

$$A = \theta(r_{max}^2 - r_{min}^2) \tag{A.26}$$

avec

$$\theta = \cos^{-1}(< \vec{d_1} \cdot \vec{d_2} > \tag{A.27}$$

Échantillonnage uniforme

Si ε_1 et ε_2 sont deux valeurs tirées selon une loi uniforme dans l'interval $[0,1]$, alors il est possible de définir une position p selon une loi uniforme sur un anneau

$$\begin{aligned} \phi &= 2\varepsilon_1\pi \\ a &= \sqrt{\varepsilon_2(R^2 - r^2) + r^2} \\ p &= 0 + a(\cos(\phi)\vec{x} + \sin(\phi)\vec{y}) \end{aligned} \tag{A.28}$$

ou sur une portion d'anneau comme cela a été défini ci-dessus

$$\begin{aligned} \phi &= \varepsilon_1\theta \\ X &= \cos(\phi)d_{1x} - \sin(\phi)d_{1y} \\ Y &= \cos(\phi)d_{1y} + \sin(\phi)d_{1x} \\ d &= \sqrt{(r_{max}^2 - r_{min}^2)\varepsilon_2 + r_min^2} \\ p &= 0 + d(X\vec{x} + Y\vec{y}) \end{aligned} \tag{A.29}$$

Subdivision uniforme

L'anneau est toujours subdivisé en quatre quadrants selon l'angle, afin de réduire la présence du trou central.

La subdivision d'une portion d'anneau se fait comme pour un "quadtree". Il est subdivisé en quatre fils d'aire équivalente. Pour cela, la portion est coupée en deux pour le paramètre angulaire $\theta' = 0.5 * \theta$, le rayon au centre r' est choisie comme suit :

$$r' = \sqrt{\frac{r_{max}^2 + r_{rmin}^2}{2}} \tag{A.30}$$

B

Volume d'interaction interactif

Afin de limiter la mémoir nécessaire pour le calcul de mise à jour incémentale dans le cas diffus (cf. Chap. 5), nous avons suggéré l'emploi des volumes d'interactions ("shafts") dynamiques. Le test cherche toujours ici à déterminer l'interaction de la boite englobante alignée sur les axes b avec le volume d'interaction entre par les boites b_1 et b_2. Dans une utilisation standard, les plans du volume d'interaction sont construits, puis le test est effectué. La procédure suivante que nous avons choisie permet de limiter au maximun la construction des plans de séparation du volume.

Dans un premier temps, il est possible d'éliminer les cas simples :

1. $b \bigcap B = \emptyset$:
   b est à l'extérieur de la boite englobante B de b_1 et de b_2 . Ainsi b est en dehors du volume d'interaction.
2. $(b \bigcap b_1 = b) \bigvee (b \bigcap b_2 = b)$:
   b est à l'intérieur de b_1 ou de b_2. Ainsi b est à l'intérieur du volume d'interaction.
3. $(b \bigcap b_1 \neq \emptyset) \bigvee (b \bigcap b_2 \neq \emptyset)$:
   b intersecte b_1 ou b_2. Ainsi b intersecte le volume d'interaction.

Par la suite, il faut construire les plans tout en les utilisant pour tester si b est en dehors du volume :

Pour chaque plan P potentiel

construire P, normal vers l'extérieur du volume

tester si b est au dessus de P selon la normal de P

Dans le cas d'une réponse positive, b est en dehors du volume d'interaction.

En sortant de cette boucle, nous savons avec certitude que b n'est pas à l'exterieur du volume d'interaction. Connaissant les plans délimitant ce volume, nous testons si b est cette fois-ci en dessous de tous les plans P, ce qui voudrait dire que b est à l'intérieur du volume.

Dans le cas contraire, il est possible d'affirmer que *b* intersecte le volume d'interaction.

C

Description des librairies

Dans cette annexe, nous allons décrire l'ensemble des librairies qui compose notre système d'illumination globale. La description complète de ces librairies est disponible en ligne[1]. Nous allons ici présenter les différentes librairies et leurs rôles.

Les objets de bases

Fichier	Objets	Librairie
Vect2.h	Vect2,Dir2 point et direction 2D	*Vect2*
Vect3.h	Vect3,Dir3 point et direction 3D	*Vect3*
Quaternion.h	Quaternion	
Spectrum.h	Spectrum,RGBColor Représentation spectrale de la lumière Transformations en couleur	*Spectrum*

TAB. C.1 : Classes de bases

Géométrie : *GeomObj*

Cette librairie est composée d'une seule hiérarchie de classes dont la racine est la classe *GeometricObj*. Tout objet dérivant de cette classe répondant aux besoins des méthodes d'éclairage global. Plus de détail se trouve dans la table Tab. C.2

[1] `http ://www-imagis.imag.fr/Xavier.Granier/GIS/WWW/index.html`

Fichier	Objets
GeomObj.h	GeometricObject AxisAlignedBox
Surface.h	Surface PolygonalPlane
Polygon.h	Polygon Triangle,Quadrilatere,NPolygon
Sphere.h	Sphere ArcSphere
Cylinder.h	Cylinder ArcCylinder
Cone.h	Cone ArcCone
Ring.h	Ring ArcRing
Point.h	PointObject

TAB. C.2 : Librairie géométrique

Propriétés matérielles : MatPro

Cette librairie est composée d'une seule hiérarchie de classes dont la racine est la classe *MaterialProperties*. Tout objet dérivant de cette classe répondant aux besoins de la plupart des méthodes d'éclairage global. Nous disposons ici de matériaux diffus constants (*DiffuseMaterial*) ou texturés (*TextureMaterial*), de matériaux de types Phong réflecteurs (*PhongMaterial*) ou transparent (*PhongTransparentMaterial*), de matériaux avec des réponses spéculaires de type Dirac réflecteurs (*MirrorMaterial*) ou transparent (*DiracTransparentMaterial*).

Gestion de la hiérarchie : *HierObj*

Cette librairie est aussi composée d'une seule hiérarchie de classe dont la racine est *HierarchicalObject*. C'est á ce niveau que se fait l'association entre la géométrie et les propriétés matérielles.

Gestion des Entrées/Sorties de fichiers : *FileIO*

Cette librairie est charger de gérer les entrées/sorties de fichier de description de scènes. Elle est chargée de choisir la bonne méthode en fonction de l'extension. Ils y a deux types de fichiers en entrée implémentés, Mgf (*MgfParser*) et Vrml97/2.0 (*VrmlParser*).

Regroupement : *Clusterizer*

Seule la méthode décrite dans le chapitre Chap. 3 a été implémentée. Elle prend en entrée un groupe racine d'une hiérarchie déjà existante, et la modifie.

Accélération du tracé de rayons : *RayEngine*

Cette librairie contient des méthodes de tracé de rayon entre deux éléments, et contient notamment la construction de la liste des occultants possibles. Cette librairie est principalement utilisé pour le calcul de visibilité. Elle est constituée d'une classe principale *RayEngine*.

Calculs de visibilité : *VisEngine*

Cette librairie contient toutes les méthodes de calcul de visibilité que cela soit entre deux points ou entre deux éléments, avec structure d'accélération ou pas. Elle est principalement basée sur la librairie de tracé de rayon. Elle est constituée d'une classe principale *VisEngine*.

Radiosité hiérarchique : *HRC*

Cette librairie est composée principalement de trois hiérarchies de classe principales. La plus importante est bien sûr celle des éléments de radiosité hiérarchique *HRCelement*, basé sur la librairie de gestion de la hiérarchie, mais ajoutant toutes les informations nécessaires à un tel algorithme : radiosité, irradiance, liste de liens, etc...

La deuxième hiérarchie de classe est celle des liens *Link*. Enfin, la dernière classe important est celle gérant une itération *HRCIterator*. Une itération fait appel à trois classes, une pour le raffinement *Refiner*, une pour le transfert d'énergie *Transfer*, et la dernière pour le mise à jour hiérarchique *PushPull*.

Méthode Unifiée : *HRCPT*

Cette librairie est principalement basée sur la précédente. Elle contient des méthodes et des classe spécifiques pour l'intégration du tracé de particules à une méthode radiosité hiérarchique (notre méthode unifiée).

Reconstruction finale : *FinalGather*

Cette librairie contient les méthodes nécessaires à une étape de reconstruction finale. Elle est basée sur une seule classe *FinalGather*.

Table des matières

1 Introduction **1**

2 État de l'art **7**

1 Principes généraux 7

1.1 Ce que l'on perçoit 7

1.2 Équation du rendu : équilibre lumineux 8

1.3 Définition d'un flux 9

1.4 Séparation des types de réflexion 10

1.5 Hypothèse diffuse : Nouvelle formulation 11

2 Les méthodes déterministes 12

2.1 Discrétisation : hypothèse diffuse 13

2.2 Facteur de forme et visibilité 14

2.3 Radiosité progressive 15

2.4 Radiosité hiérarchique avec regroupement 15

Principes 16

Algorithme 17

Le regroupement ("clustering") 18

Le raffinement 19

2.5 Reconstruction finale "Final Gather" 19

2.6 État des approches déterministes 20

3 Les méthodes stochastiques 21

3.1 Méthodes de Monte-Carlo 21

3.2 Application au problème de l'éclairage 23

Lancer de chemins 23

Lancer de particules 24

Méthodes bidirectionnelles 26

Méthodes de Metropolis 26

3.3 Les méthodes hybrides . . . 27
Méthodes en deux étapes . . . 27
Méthodes stochastiques sur maillage . . . 27
Méthodes en plusieurs étapes . . . 27
3.4 État des approches stochastiques . . . 28
4 Qualité d'un éclairage . . . 28
4.1 Contrôle de l'erreur commise . . . 28
Modélisation réaliste de la scène . . . 29
Rendre toutes les réflexions . . . 30
Bonne représentation de toute l'échelle de variation . . . 30
Gestion de la complexité . . . 30
4.2 Contrôle de la qualité visuelle . . . 30
4.3 Automatisation . . . 31

3 Plateforme d'expérimentation 33
1 Architecture logicielle : G.I.S. . . . 34
1.1 Modules généraux . . . 34
1.2 Module de radiosité hiérarchique . . . 35
2 Traitement hiérarchique . . . 36
2.1 Regroupements . . . 36
2.2 Calculs sur des groupes . . . 39
2.3 Traitement des objets . . . 39
3 Raffinement . . . 40
3.1 Oracle de raffinement . . . 40
3.2 Calcul de visibilité et du facteur de forme . . . 42
4 Lissage . . . 43
4.1 Utilisation d'une grille . . . 43
4.2 Structure de point . . . 44
4.3 Suivi des ombres . . . 44
5 Conclusion . . . 44

4 Nouvelle méthode unifiée 47
1 Introduction . . . 47
2 Aperçu général de la méthode . . . 48
2.1 Combiner les approches et répartir les calculs . . . 49
2.2 Utilisation de la structure hiérarchique . . . 49
2.3 Une itération . . . 50
Raffinement . . . 51
Transfert d'énergie . . . 51
"Push/Pull" et placement des particules . . . 51
3 Intégration des deux algorithmes . . . 52
3.1 Émission de particules . . . 53
3.2 Placement hiérarchique des particules . . . 55
3.3 Intégration des particules à la solution de radiosité . . . 56
4 Accélération de la simulation . . . 57
5 Reconstruction et rendu . . . 58

5.1 Lissage . . . 58
5.2 Ajout de la composante dépendante du point de vue . . . 58
6 Résultats . . . 59
6.1 Rendu interactif pour les scènes simples . . . 59
6.2 Contrôle de qualité . . . 59
6.3 Scènes complexes et éclairage indirect . . . 60
6.4 Comparaison informelle avec un lancer de particules . . . 60
6.5 Visualisation interactive et “Render Cache” *[WDP99]* . . . 61
7 Conclusion . . . 61
Limitations . . . 62

5 Mise à jour incrémentale 65
1 Introduction . . . 65
1.1 Motivations . . . 65
1.2 Positionnement de notre méthode . . . 66
Radiosité hiérarchique dynamique . . . 68
2 Reconstruction des caustiques par textures . . . 68
2.1 Choix de la représentation . . . 71
2.2 Déterminer la résolution de la texture . . . 72
2.3 Filtrage . . . 73
2.4 Restriction . . . 74
3 Lancer de particules incrémental . . . 75
3.1 Octree dynamique . . . 76
3.2 Mise à jour incrémental des particules . . . 76
3.3 Mise à jour interactive avec dégradation de qualité . . . 77
4 Cas diffus . . . 78
5 Affichage interactif de la solution . . . 79
5.1 Affichage des textures de caustique . . . 79
5.2 Paramétrisations des textures de caustique . . . 81
6 Résultats . . . 81
7 Conclusion . . . 85

6 Reconstruction finale 89
1 Présentation générale . . . 90
2 Reconstruction de la composante diffuse . . . 90
2.1 Initialisation d’un élément . . . 91
Sur les éléments . . . 92
Sur les sommets . . . 92
2.2 Calcul de la valeur d’irradiance . . . 93
Avec les liens . . . 93
Avec les sommets . . . 94
3 Reconstruction et intégration des caustiques . . . 95
3.1 Utilisation des particules existantes . . . 95
3.2 Réémission de particules . . . 96
4 Composante dépendante du point de vue . . . 97
4.1 Utilisation d’un lancer de rayons . . . 98

4.2 Séparation direct/indirect 99
4.3 Utilisation des directions des particules 99
4.4 Combinaisons avec des rayons 100
5 Premiers résultats 101
5.1 Diffus seul 101
5.2 Caustiques et Composante dépendante du point de vue 102
5.3 Scènes complexes 103
6 Conclusion 105

7 Conclusion **107**
Résumé des contributions 107
Travaux futurs 108
Pour conclure 109

Annexes 109

A Paramétrisation des objets **113**
1 Sphère et portion de sphère 113
Choix de paramétrisation 113
Calcul de l'aire des portions de sphères 114
Échantillonnage uniforme 114
2 Cône et portion de cône 116
Choix de paramétrisation 116
Calcul des aires 117
Échantillonnage uniforme 117
Subdivision uniforme 118
3 Cylindre et portion de cylindre 119
Choix de paramétrisation 119
Calcul de l'aire d'une portion de cylindre 120
Échantillonnage uniforme 120
Subdivision uniforme 120
4 Anneau et portion d'anneau 122
Choix de paramétrisation 122
Calcul des aires 123
Échantillonnage uniforme 123
Subdivision uniforme 123

B Volume d'interaction interactif **125**

C Description des librairies **127**
Les objets de bases 127
Géométrie : *GeomObj* 127
Propriétés matérielles : MatPro 128
Gestion de la hiérarchie : *HierObj* 128
Gestion des Entrées/Sorties de fichiers : *FileIO* 128

Regroupement : *Clusterizer* 129
Accélération du tracé de rayons : *RayEngine* 129
Calculs de visibilité : *VisEngine* 129
Radiosité hiérarchique : *HRC* 129
Méthode Unifiée : *HRCPT* 129
Reconstruction finale : *FinalGather* 129

Table des figures

1.1 Images tirées de trois films récents 2
1.2 Composition des effets lumineux 5

2.1 Angle solide 9
2.2 Différent types de propriété de réflexion 10
2.3 Des chemins lumineux 11
2.4 Algorithme de radiosité progressive (cas de l'émission progressive) 15
2.5 Quatre échanges à des niveaux différents 16
2.6 Détail de la mise à jour hiérarchique 17
2.7 Émission d'un rayon à partir du point de vue 24
2.8 Comparaison de deux méthodes stochastiques 26
2.9 Les étapes de la simulation d'éclairage (adapté de *[ATS94]*) 29

3.1 Architecture du système d'éclairage global 35
3.2 Géométrie du test de regroupement 36
3.3 Première amélioration du regroupement 37
3.4 Seconde amélioration du regroupement 38
3.5 Descente des liens sur les objets 39
3.6 Oracle de raffinement 41
3.7 Construction de la grille 43
3.8 Variation du nombres d'itérations de lissage sur une grille 44
3.9 Suivie des ombres 45

4.1 Deux images extraites d'une session interactive 49
4.2 Espace des directions du aux liens 50
4.3 Émission et propagation des particules 52
4.4 Placement hiérarchique des particules 56
4.5 Utilisation de l'information de visibilité 57

4.6 Utilisation du Render Cache *[WDP99]* 58
4.7 Variation du paramètre ϕ_{ct} 59
4.8 Deux vues d'une même scène. 60
4.9 La scène Soda Shoppe scène avec un éclairage indirect 61
4.10 Deux images d'une session interactive avec le "Render Cache" *[WDP99]*. . . . 62
4.11 Promenade interactive par rendu OpenGL 63

5.1 Quelques résultats sur une scène l'intérieur 67
5.2 Une commode avec des caustiques 69
5.3 Construction de l'histogramme 71
5.4 Trois spirales de reconstruction 73
5.5 Détection de la zone de restriction 74
5.6 Étude comparative de la restriction 75
5.7 Description de l'octree dynamique 77
5.8 Le mouvement capté par un octree 78
5.9 Déplacement du pot sur la commode 80
5.10 Scène d'intérieur 82
5.11 Quatre postions du récipient sur la commode de la chambre 83
5.12 Scène d'extérieur ($47\,000$ polygones) 85
5.13 Quatre positions de la voiture dans la scène d'extérieur 86

6.1 Aperçu de l'algorithme de reconstruction final 91
6.2 Reconstruction diffuse d'un bureau 94
6.3 Émission d'un rayon à partir du point de vue 98
6.4 Reconstruction diffuse d'un salon 101
6.5 Scène avec deux pièces en éclairage indirecte. 102
6.6 Salle à manger en éclairage directe. 103
6.7 Test du critère de réémission. 103
6.8 Reconstruction diffuse du "Soda Shoppe" dans un éclairage indirecte. 104
6.9 Deux points de vue de l'intérieur d'un bateau 104
6.10 Scène de couloir. 106

A.1 Représentation d'une portion de sphère 114
A.2 Représentation d'une portion de cône 116
A.3 Représentation d'une portion de cône 118
A.4 Représentation d'une portion de cylindre 119
A.5 Subdivision d'une portion de cylindre 120
A.6 Représentation d'une portion d'anneau 122

Liste des tableaux

2.1 Les différentes grandeurs 12
2.2 Extension du facteur de forme *[Sil95]* 19

5.1 Résultats de deux sessions incrémentales 84

C.1 Classes de bases 127
C.2 Librairie géométrique 128

Bibliographie

[AH93] Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Proc. SIGGRAPH'93*, Computer Graphics Proceedings, Annual Conference Series, pages 155–162, 1993.

[Arv86] James R. Arvo. Backward ray tracing. In *ACM SIGGRAPH '86 Course Notes - Developments in Ray Tracing*, volume 12, Aôut 1986.

[Arv95] James Arvo. Stratified sampling of spherical triangles. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '95*, Computer Graphics Proceedings, Annual Conference Series, pages 437–438. ACM SIGGRAPH, ACM Press, Août 1995.

[ATS94] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 75–84. ACM SIGGRAPH, ACM Press, Juillet 1994.

[BA97] Adrina W. Bowman and Adelchi Azzalini. *Applied Smoothing Techniques for Data Analisys*. Clarendon Press - Oxford, 1997.

[BDW98] Ph. Bekaert, Ph. Dutre, and Y. D. Willems. Final radiosity gather step using a monte carlo technique with optimal importance sampling. Technical Report CW275, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, November 1998.

[Bek99] Philippe Bekaert. *Hierarchical and Stochastic Algorithms for Radiosity*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 1999.

[BGB97] Rui Bastos, Michael Goslin, and Norman I. Badler. Efficient rendering of radiosity using texture and bicubic interpolation. In *1997 Symposium on Interactive 3D Graphics*, pages 71–74. ACM SIGGRAPH, April 1997.

[BNN+98] Philippe Bekaert, Lázló Neumann, Atilla Neumann, Mateu Sbert, and Yves D. Willems. Hierarchical monte-carlo radiosity. In George Drettakis and Nelson Max, editors, *Eurographics Workshop on Rendering*. Eurographics, 1998.

[BRW89] Daniel R. Baum, Holly E. Rushmeier, and James M. Winget. Improving radiosity solutions through the use of analytically determined form-factors. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23 of *Computer Graphics Proceedings, Annual Conference Series*, pages 325–334. ACM SIGGRAPH, ACM Press, Juillet 1989.

[CCWG88] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22 of *Computer Graphics Proceedings, Annual Conference Series*, pages 75–84. ACM SIGGRAPH, ACM Press, Août 1988.

[CG85] Michael F. Cohen and Donald P. Greenberg. The hemi-cube : A radiosity solution for complex environnement. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19 of *Computer Graphics Proceedings, Annual Conference Series*, pages 31–40. ACM SIGGRAPH, ACM Press, Août 1985.

[Che90] Shenchang Eric Chen. Incremental radiosity : An extension of progressive radiosity to an interactive image synthesis system. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24 of *Computer Graphics Proceedings, Annual Conference Series*, pages 135–144. ACM SIGGRAPH, ACM Press, Août 1990.

[CLSS97] Per H. Christensen, Dani Lischinski, Eric J. Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1) :3–33, Janvier 1997.

[Coo86] Robert L. Cook. Practical aspects of distributed ray tracing. In *ACM SIGGRAPH '86 Developments in Ray Tracing seminar notes*, August 1986.

[CPC84] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 137–145, Juillet 1984.

[CRMT91] Shenchang Eric Chen, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass method for global illumination. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25 of *Computer Graphics Proceedings, Annual Conference Series*, pages 165–174. ACM SIGGRAPH, ACM Press, Juillet 1991.

[DB97] Paul J. Diefenbach and Norman I. Badler. Multi-pass pipeline rendering : Realism for dynamic environments. *1997 Symposium on Interactive 3D Graphics*, pages 59–70, Avril 1997.

[DBG99] Reynald Dumont, Kadi Bouatouch, and Philippe Gosselin. A progressive algorithm for three point transport. *Computer Graphics Forum*, 18(1) :41–56, Mars 1999.

[DDP97] Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton : A powerful and multi-purpose global visibility tool. In *ACM SIGGRAPH 97*, Août 1997. http ://www-imagis.imag.fr/Publications/1997/DDP97b.

[DDP99] Frédo Durand, George Drettakis, and Claude Puech. Fast and accurate hierarchical radiosity using global visibility. *ACM Transactions on Graphics*, 18(2) :128–170, Avril 1999. http ://www.acm.org :80/pubs/citations/journals/tog/1999-18-2/p128-durand/.

[DF94] George Drettakis and Eugene Fiume. A fast shadow algorithm for area light sources using backprojection. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 75–84. ACM SIGGRAPH, ACM Press, Juillet 1994.

[DS96] George Drettakis and François X. Sillion. Accurate visibility and meshing calculations for hierarchical radiosity. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 269–278, New York City, NY, Juin 1996. Eurographics, Springer Wein.

[DS97] George Drettakis and François X. Sillion. Interactive update of global illumination using a line-space hierarchy. In Turner Whitted, editor, *Computer Graphics Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 57–64. ACM SIGGRAPH, ACM Press, Août 1997.

[DSSD97] Katja Daubert, Hartmut Schirmacher, François X. Sillion, and George Drettakis. Hierarchical lighting simulation for outdoor scenes. In Julie Dorsey and Philipp Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 229–238, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.

[Dum96] Reynald Dumont. *Optimization du calcul d'illumination globale de scènes intérieures/extérieures : niveaux de détails, structuration et réflectance bidirectionnelle quelconque*. PhD thesis, Katholieke universiteit leuven, Heverlee - Belgium, Septembre 1996.

[Dur99] Frédo Durand. *Visibilité tridimensionnelle : Etude analytique et applications*. PhD thesis, Université Joseph Fourier (Grenoble), Juillet 1999.

[Dut] Philip Dutré. Gobal illumination compendium. Technical report, Progran of Computer Graphics, Cornell University. http ://www.graphics.cornell.edu/ phil/GI.

[Dut96] Philip Dutre. *Mathematical frameworks and Monte Carlo alogorithms for global illumination in Computer Graphics*. PhD thesis, Katholieke universiteit leuven, Heverlee - Belgium, Septembre 1996.

[FP93] Martin Feda and Werner Purgathofer. Progressive ray refinement for monte carlo radiosity. In *Fourth Eurographics Workshop on Rendering*, number Series EG 93 RW, pages 15–26, Paris, France, June 1993.

[GD99] Xavier Granier and George Drettakis. Controlling memory consumption of hierarchical radiosity with clustering. In *Graphics Interface*, pages 58–65, Juin 1999. http ://www.dgp.toronto.edu/gi99/papers/146.

[GD01] X. Granier and G. Drettakis. Incremental updates for rapid glossy global illumination. In *Computer Graphics Forum (Proceedings of Eurographics 2001)*, volume 20, September 2001. To appear.

[GDW00] Xavier Granier, George Drettakis, and Bruce Walter. Fast global illumination including specular effects. In B. Peroche and H. Rushmeier, editors, *Rendering*

Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering), pages 47–58, New York, NY, 2000. Springer Wien.

[GH96] Simon Gibson and R. J. Hubbold. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5) :297–310, Décembre 1996.

[GS96] Sherif Ghal and A. James Stewart. A complete treatment of d1 discontinuities in a discontinuity mesh. In Wayne A. Davis and Richard Bartels, editors, *Graphics Interface '96*, pages 122–131. Canadian Information Processing Society, Canadian Human-Computer Communications Society, Mai 1996. http ://www.dgp.toronto.edu/gi/gi96/proceedings/papers/GhaliStewart.

[GSG90] David W. George, Francois X. Sillion, and Donald P. Greenberg. Radiosity Redistribution for Dynamic Environments. *IEEE Computer Graphics and Applications*, 10(4) :26–34, July 1990.

[GTGB84] Cindy M. Goral, Kenneth K. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics Proceedings*, volume 18 of *Computer Graphics Proceedings, Annual Conference Series*, pages 213–222. ACM SIGGRAPH, ACM Press, Juillet 1984.

[HCA00] Nicolas Holzschuch, François Cuny, and Laurent Alonso. Wavelet radiosity on arbitrary planar surfaces. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 161–172, New York, NY, 2000. Springer Wien.

[HDSD99] J. M. Hasenfratz, C. Damez, F. X. 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.

[Hec90] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In Forest Baskett, editor, *Computer Graphics Proceedings*, volume 24 of *Computer Graphics Proceedings, Annual Conference Series*, pages 145–154. ACM SIGGRAPH, ACM Press, Août 1990.

[Hec91] Paul S. Heckbert. *Simulating Global Illumination Using Adaptative Meshing*. PhD thesis, University of California, Berkeley, Mai 1991.

[HH84] Paul S. Heckbert and Pat Hanrahan. Beam tracing polygonal objects. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH'84 Proceedings)*, volume 18 of *Computer Graphics Proceedings, Annual Conference Series*, pages 119–127. ACM SIGGRAPH, ACM Press, Juillet 1984.

[How82] J. R. Howell. *A Catalog of Radiation Configuration Factors*. McGraw-Hill, New York, 1982. http ://www.me.utexas.edu/ howell/tablecon.html.

[HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In Thomas W. Sederberg, editor, *Computer Graphics Proceedings*, volume 25 of *Computer Graphics Proceedings, Annual Conference Series*, pages 197–206. ACM Press, Juillet 1991.

[HW94] Eric A. Haines and John R. Wallace. Shaft culling for efficient ray-traced radiosity. In P. Brunet and F. W. Jansen, editors, *Eurographics Workshop on Rendering*, New York, NY, 1994. Eurographics, Springer-Verlag.

[ICG86] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. In David C. Evans and Russell J. Athay, editors, *Computer Graphics Proceedings*, volume 20 of *Computer Graphics Proceedings, Annual Conference Series*, pages 133–142. ACM SIGGRAPH, ACM Press, Août 1986.

[JC94] Henrik Wann Jensen and Niels Jørgen Christensen. Photon maps in bidirectional monte carlo ray tracing of complex objects. In *Computer Graphics*, volume 19 of *Eurographics Proceedings, Annual Conference Series*, pages 215–224, Mars 1994.

[Jen96] Henrik Wann Jensen. Global illumination using photon maps. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Workshop on Rendering*, pages 21–30, New York City, NY, Juin 1996. Eurographics, Springer Wien.

[Jen01] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, LTD., Natick, 2001.

[Kaj86] James T. Kajiya. The rendering equation. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20 of *Computer Graphics Proceedings, Annual Conference Series*, pages 143–150. ACM SIGGRAPH, ACM Press, Août 1986.

[Kok93] Arjan J. F. Kok. Grouping of Patches in Progressive Radiosity. In *Fourth Eurographics Workshop on Rendering*, number Series EG 93 RW, pages 221–232, Paris, France, June 1993.

[KSS97] Leif Kobbelt, Marc Stamminger, and Hans-Peter Seidel. Using subdivision on hierarchical data to reconstruct radiosity distribution. In D. Fellner and L. Szirmay-Kalos, editors, *Computer Graphics Forum*, volume 16 of *Eurographics Proceedings, Annual Conference Series*, pages 347–355, Budapest, Hungary, Septembre 1997. Eurographics, Blackwell Publishers.

[LF96] Robert R. Lewis and Alain Fournier. Light-Driven Global Illumination with a Wavelet Representation. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 11–20, New York, NY, 1996. Springer-Verlag/Wien.

[LSG94] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. In Andrew Glassner, editor, *Computer Graphics Proceedings*, volume 28 of *Computer Graphics Proceedings, Annual Conference Series*, pages 67–74. ACM SIGGRAPH, ACM Press, Juillet 1994.

[LTG92] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Discontinuity Meshing for Accurate Radiosity. *IEEE Computer Graphics and Applications*, 12(6) :25–39, Novembre 1992.

[LTG93a] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining Hierarchical Radiosity and Discontinuity Meshing. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 199–208, 1993.

[LTG93b] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Procee-*

dings, Computer Graphics Proceedings, Annual Conference Series, pages 199–208. ACM SIGGRAPH, ACM Press, 1993.

[LW93] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In H. P. Santo, editor, *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, December 1993.

[LW94] Eric P. Lafortune and Yves D. Willems. Using the modified phong reflectance model for physically based rendering. Technical Report CW 197, Departement of Computing Science, K.U. Leuven, Novembre 1994.

[MBG86] Gregory M. Maxwell, Michael J. Bailey, and Victor W. Goldschmidt. Calculations of the radiation configuration factor using ray casting. *Computer-Aided Design*, 18(7) :371–379, Septembre 1986.

[McC99] Michael D. McCool. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Transactions on Graphics*, 18(2) :171–194, April 1999. http ://www.acm.org :80/pubs/citations/journals/tog/1999-18-2/p171-mccool/.

[MK94] Karol Myszkowski and Tosiyasu L. Kunii. Texture Mapping as an Alternative for Meshing During Walkthrough Animation. In *Fifth Eurographics Workshop on Rendering*, pages 375–388, Darmstadt, Germany, June 1994.

[Neu95] Laszlo Neumann. Monte carlo radiosity. *Computing*, 55(1) :23–42, 1995.

[Ng96] Adelene Ng. A comparison of four visibility acceleration techniques for radiosity. *The Visual Computer*, 12(6) :307–316, 1996.

[NNB97] Lázló Neumann, Atilla Neumann, and Philippe Bekaert. Radiosity with well distributed ray sets. In D. Fellner and L. Szirmay-Kalos, editors, *Computer Graphics Forum*. Eurographics, 1997.

[NRH+77] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, , and T. Limperis. *Geometrical Considerations and Nomenclature for Reflectance*. National Bureau of Standards, Washington, October 1977.

[OF99] Marc Ouellette and Eugene Fiume. Approximating the location of integrand discontinuities for penumbral illumination computation with area light sources. In Dani Lischinski and Greg Ward Larson, editors, *10th Eurographics Rendering Workshop*. Springer Wein, Eurographics, Juin 1999.

[Pho75] Bui-Tuong Phong. Illumination for computer generated pictures. *CACM June 1975*, 18(6) :311–317, 1975.

[PM92] Sumanta 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, Bristol, UK, May 1992.

[PMS+99] Steven Parker, William Martin, Peter-Pike J. Sloan, Peter Shirley, Brian Smits, and Charles Hansen. Interactive ray tracing. *1999 ACM Symposium on Interactive 3D Graphics*, pages 119–126, Avril 1999.

[PS89] J. Painter and K. Sloan. Antialiased ray tracing by adaptive progressive refinement. In *Conference proceedings on Computer graphics*, pages 281 – 288, 1989.

[RR99] F. Rousselle and C. Renaud. Group accelerated shooting methods for radiosity. In *Rendering Techniques '99*, pages 77–88, New York, NY, 1999. Springer Wien.

[RSH00] Erik Reinhard, Brian Smits, and Chuck Hansen. Dynamic acceleration structures for interactive ray tracing. In *Rendering Techniques 2000 (Proceedings of the 11th Eurographics Workshop on Rendering)*, pages 299–306, Juin 2000.

[SAG94] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In Andrew Glassner, editor, *Computer Graphics Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442. ACM SIGGRAPH, ACM Press, Juillet 1994.

[SAWG91] François X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In *Computer Graphics*, volume 25 of *Computer Graphics Proceedings, Annual Conference Series*. ACM, Juillet 1991.

[SB97] W. Stürzlinger and R. Bastos. Interactive rendering of globally illuminated glossy scenes. In *8th EG Workshop on Rendering, "Rendering Techniques '97"*, pages 93–102. Springer Wien, Juin 1997.

[SC78] E. M. Sparrow and R. D. Cess. *Radiation Heat Transfer*. Hemisphere Publishing Corporation, Washington, 1978.

[Sch97] Stephan Schaefer. Hierarchical radiosity on curved surfaces. In Julie Dorsey and Philipp Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 187–192, New York, NY, 1997. Springer Wien.

[SD95] François X. Sillion and George Drettakis. Feature-based control of visibility error : A multi-resolution clustering algorithm for global illumination. In Robert Cook, editor, *Computer Graphics Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 145–152. ACM SIGGRAPH, ACM Press, Août 1995.

[SDS95] François X. Sillion, George Drettakis, and Cyril Soler. A clustering algorithm for radiance calculation in general environments. In P. M. Hanrahan and W. Purgathofer, editors, *Eurographics Workshop on Rendering*, pages 196–205, New York, NY, 1995. Eurographics, Springer-Verlag.

[SH81] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corporation, Washington, D.C., 1981.

[Sha97] Erin Shaw. Hierarchical radiosity for dynamic environments. *Computer Graphics Forum*, 16(2) :107–118, 1997.

[Shi90] Peter Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. In *Graphics Interface*, pages 205–212, 1990.

[Shi91] Peter Shirley. Radiosity via ray tracing. In James Arvo, editor, *Graphics Gems II*, pages 306–310. Academic Press Professional, Boston, MA, 1991.

[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) :240–254, 1995.

[SLD92] David Salesin, Daniel Lischinski, and Tony DeRose. Reconstructing illumination functions with selected discontinuities. In *Third Eurographics Workshop on Rendering*, pages 99–112, Mai 1992.

[Sol98] Cyril Soler. *Représentations hiérarchiques de la visibilité pour le contrôle de l'erreur en simulation de l'éclairage*. PhD thesis, Université Joseph Fourier (Grenoble), Décembre 1998.

[SP89] François X. Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. In *Computer Graphics Proceedings*, volume 23 of *Computer Graphics Proceedings, Annual Conference Series*, pages 335–344. ACM SIGGRAPH, Juillet 1989.

[SP94] François X. Sillion and Claude Puech. *Radiosity & Global Illumination*. Morgan Kauffmann, San Fransisco, California, 1994.

[SP99] F. Schoeffel and A. Pomi. Reducing memory requirements for interactive radiosity using movement prediction. In *Rendering Techniques '99*, pages 225–234, New York, NY, 1999. Springer Wien.

[Spa63] E. M. Sparrow. A new and simpler formulation for radiative angle factors. *ASME Journal of Heat Transfer*, 85(2) :81–88, Mai 1963.

[SPP95] Mateu Sbert, Frederic Perez, and Xavier Pueyo. Global Monte Carlo : A Progressive Solution. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 231–239, New York, NY, 1995. Springer-Verlag.

[SS00] Cyril Soler and Francois X. Sillion. Texture-based visibility for efficient lighting simulation. *ACM Transactions on Graphics*, 19(4), October 2000. To appear.

[SSG+99] Marc Stamminger, Annette Scheel, Xavier Granier, Frederic Perez-Cazorla, George Drettakis, and François X. Sillion. Efficient glossy global illumination with interactive viewing. In *Graphics Interface*, pages 50–57, Juin 1999. http ://www.dgp.toronto.edu/gi99/papers/145.

[SSG+00] Marc Stamminger, Annette Scheel, Xavier Granier, Frederic Perez-Cazorla, George Drettakis, and François X. Sillion. Efficient glossy global illumination with interactive viewing. *Computer Graphic Forum*, 19(1) :13–25, Mars 2000.

[SSS97] Marc Stamminger, Philipp Slusallek, and Hans-Peter Seidel. Bounded radiosity - illumination on general surfaces and clusters. In D. Fellner and L. Szirmay-Kalos, editors, *Computer Graphics Forum*, volume 16 of *Eurographics Proceedings, Annual Conference Series*. Eurographics, 1997.

[SSS98] Marc Stamminger, Philipp Slusallek, and Hans-Peter Seidel. Three point clustering for radiance computations. In *9th EG Workshop on Rendering, "Rendering Techniques '98"*, pages 211–222. Springer Wien, 1998.

[SSS01] Annette Scheel, M. Stamminger, and Hans-Peter Seidel. Thrifty final gather for radiosity. In *12th EG Workshop on Rendering, "Rendering Techniques 2001"*. Springer Wien, 2001. To appear.

[Stu94] W. Sturzlinger. Adaptive mesh refinement with discontinuities for the radiosity method. In *Fifth Eurographics Workshop on Rendering*, pages 239–248, Juin 1994.

[SW99] Frank Suykens and Yves D. Willems. Combining bidirectional path tracing and multipass rendering. In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)*, pages 265–272, Plzen-Borey, Czech Republic, February 1999. University of West Bohemia.

[SW00] Frank Suykens and Yves D. Willems. Density control for photon maps. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 23–34, New York, NY, 2000. Springer Wien.

[TH94] Seth Teller and Pat Hanrahan. Global visibility algorithms for illumination computations. In *Computer Graphics*, pages 443–450. ACM, 1994.

[TNSP98] Robert F. Tobler, Laszlo Neumann, Mateu Sbert, and Werner Purgathofer. A new form factor analogy and its application to stochastic global illumination algorithms. In G. Drettakis and N. Max, editors, *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pages 35–44, New York, NY, 1998. Springer Wien.

[TS01] Jérémie Turbet and François X. Sillion. A graph process for easy design of refiners in hierarchical radiosity. In *WSCG (Winter School of Computer Graphics)*, 2001. http ://www-imagis.imag.fr/Publications/2001/TS01.

[Tur01] Jérémie Turbet. *Études de critères de raffinement en radiosité hiérarchique*. PhD thesis, Université Joseph Fourier (Grenoble), 2001.

[TWFP97] Robert F. Tobler, Alexander Wilkie, Martin Feda, and Werner Purgathofer. A hierarchical subdivision algorithm for stochastic radiosity methods. In Julie Dorsey and Philipp Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 193–204, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.

[Vea97] Eric Veach. *Robust Monte-Carlo Methods for Light Transport Simulation*. PhD thesis, Standford University, 1997.

[VG94] Eric Veach and Leonidas J. Guibas. Bidirectional estimators for light transport. In *Eurographics Workshop on Rendering*, pages 147–162. Eurographics, Springer Wien, Juin 1994.

[VG97] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Proceedings of SIGGRAPH '97*, volume 31 of *Computer Graphics Proceedings, Annual Conference Series*, pages 65–76. ACM SIGGRAPH, ACM Press, Août 1997.

[Wal98] B. J. Walter. *Density estimation techniques for global illumination*. PhD thesis, Cornell University, 1998. http ://www.graphics.cornell.edu/pubs/1998/Wal98.html.

[War94] Gregory J. Ward. The radiance lighting simulation and rendering system. In Andrew Glassner, editor, *Proceedings of SIGGRAPH'94*, volume 21 of *Computer Graphics Proceedings, Annual Conference Series*, pages 459–472. ACM SIGGRAPH, ACM Press, Juillet 1994. http ://radsite.lbl.gov/radiance/papers/sg94.1/paper.html.

[Wat90] Mark Watt. Light-water interaction using backward beam tracing. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24 of *Computer Graphics Proceedings, Annual Conference Series*, pages 377–385, Août 1990.

[WCG87] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation : A synthesis of ray tracing and radiosity methods. In Maureen C. Stone, editor, *Computer Graphics Proceedings*, volume 21 of *Computer Graphics Proceedings, Annual Conference Series*, pages 311–320. ACM SIGGRAPH, ACM Press, Juillet 1987.

[WDP99] Bruce J. Walter, Georges Drettakis, and Steve Parker. Interactive rendering using the render cache. In *10th EG Workshop on Rendering, "Rendering Techniques'99"*. Springer Wien, Juin 1999. Granada, Spain.

[WEH89] John R. Wallace, Kells A. Elmquist, and Eric A. Haines. A ray tracing algorithm for progressive radiosity. In *Computer Graphics Proceedings*, volume 23 of *Computer Graphics Proceedings, Annual Conference Series*, pages 315–324. ACM SIGGRAPH, ACM Press, Juillet 1989.

[WH92] Gregory J. Ward and Paul Heckbert. Irradiance gradients. In Derek Paddon Alan Chalmers and François X. Sillion, editors, *Third Eurographics Workshop on Rendering (Rendering Techniques '92)*, pages 85–98, Bristol, UK, Mai 1992. Consolidation Express Bristol.

[WH97a] Andrew J. Willmott and Paul Heckbert. An empirical comparison of progressive and wavelet radiosity. In Julie Dorsey and Phillip Slusallek, editors, *Eurographics Workshop on Rendering*, pages 175–186, New York, NY, 1997. Eurographics, Springer Wien.

[WH97b] Andrew J. Willmott and Paul S. Heckbert. An empirical comparison of radiosity algorithms. Technical Report CMU-CS-97-115, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Avril 1997. http ://www.cs.cmu.edu/˜radiosity/emprad-tr.html.

[WHG99] Andrew J. Willmott, P. Heckbert, and M. Garland. Face cluster radiosity. In *Eurographics Rendering Workshop 1999*, pages 293–304, New York, NY, Juin 1999. Eurographics, Springer Wien.

[WHSG97] Bruce Walter, Philip M. Hubbard, Peter Shirley, and Donald P. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3) :217–259, Juillet 1997.

[Wie66] John A. Wiebelt. *Engineering Radiation Heat Transfer*. Holt, Rinehart and Winston, Inc, New York, 1966.

[WSBW01] Ingo Wald, Philipp Slusallek, Carsten Benthin, and Markus Wagner. Interactive distributed ray tracing of highly complex models. In *Rendering Techniques 2001 (Proceedings of the 12th Eurographics Workshop on Rendering)*, Juin 2001.

[ZSP98] Jacques Zaninetti, Xavier Serpaggi, and Bernard Peroche. A vector approach for global illumination in ray tracing. *Computer Graphics Journal (Proc. Eurographics '98)*, 17(3) :C149–C158, September 1998.

Résumé

Dans ce document, nous présentons une nouvelle approche qui par l'intégration d'une méthode de radiosité hiérarchique avec regroupement, avec une méthode de lancer de particules, permet de simuler efficacement l'ensemble des chemins lumineux.

Dans un premier temps, nous présentons une solution permettant cette intégration. Nous restreignons le lancer de particules pour les seuls échanges lumineux où cela se trouve être nécessaire. Pour cela, nous utilisons la structure de liens créée par la méthode de radiosité hiérarchique avec regroupement. Cette structure peut être considérée comme un partionnement de l'espace des échanges lumineux. Puis, nous présentons comment intégrer l'énergie due à ces particules à la solution globale. L'algorithme unifié ainsi obtenu permet une détection automatique des régions où un lancer de particules se révèle nécessaire et de plus, il permet une bonne variation entre une première solution rapide offrant une visualisation de ce que peut être un résultat final, et une solution de plus grande qualité, mais avec un temps de calcul plus élevé.

Dans un deuxième temps, nous montrons comment cette approche unifié peut s'adapter au cas dynamique. Nous introduisons pour cela une structure spatiale permettant de détecter efficacement, toujours à l'aide des liens, quelles sont les particules affectées par le déplacement d'un objet et qu'il faut donc renvoyer. Pour accélérer et améliorer le résultat dans ce cadre, nous introduisons une nouvelle méthode de reconstruction des effets lumineux dus aux particules, par l'utilisation de textures. L'algorithme ainsi présenté permet une mise à jour incrémentale rapide pour les scènes dynamiques.

Pour finir, nous présentons une méthode de reconstruction finale, qui, en extrayant les informations contenues dans une solution de notre méthode unifiée, permet d'obtenir des images de très hautes qualité, contenant l'ensemble des effets lumineux.

Abstract

In this thesis, we present a new approach which integrates hierarchical radiosity with clustering with a particle tracing method, thus simulating all global illumination phenomena.

Initially, we show how this integration is possible. We restrict particle tracing only to light exchanges for which this is necessary. To achieve this, we use the link structure created by hierarchical radiosity with clustering. This structure can be regarded as a subdivision of the space of light exchanges. We then explain how to integrate the energy of these particles into the normal radiosity solution. This unified algorithm allows automaticdetection of the regions where particle tracing is required. It also allows smooth and natural variation between a first fast solution offering initial visualization of the final result, and a higher quality solution, but with higher computation time.

The second contribution of this thesis is to show how our unified approach can deal with dynamic scenes. We introduce a spatial structure to efficiently detect which particles are affected by object motion and which particles should thus be re-emitted. This detection also uses the link structure. To accelerate and improve the result, we introduce a new reconstruction method for light effects due to particles, using textures. The algorithm thus presented allows fast incremental updates for dynamic scenes.

The third contribution is a final gather method. In this approach, we gather the information contained in a solution from our unified method, and we then use it to compute very high quality images, with all global illumination effects.