

Table des matières

Table des figures	iii
1 Introduction	1
1.1 Le sujet :	1
1.1.1 Motivation	1
1.1.2 Approche adoptée	2
1.2 Plan du rapport	2
1.3 Présentation du cadre du projet : Le projet <i>iMAGIS</i>	3
2 Etat de l'art	5
2.1 Introduction : Apport de l'illumination globale	5
2.2 Présentation générale de la radiosité	6
2.2.1 Les échanges radiatifs	6
2.2.2 Les équations de radiosité en milieu diffus	6
2.3 Discrétisation	7
2.3.1 Formulation générale	7
2.3.2 Résolution Matricielle	8
2.3.3 La radiosité progressive	8
2.4 La radiosité hiérarchique	10
2.4.1 Principes	10
2.4.2 Notations	11
2.4.3 Algorithme	12
2.4.4 Stabilité	12
2.4.5 Les clusters	13
2.4.6 Le raffinement	13
2.4.7 Conclusion	13
3 Etude de la structure de radiosité hiérarchique	14
3.1 Introduction	14
3.2 Les liens	15
3.2.1 Utilisation des liens passifs	15
3.2.2 Evolutions en nombre	16
3.2.3 Evolution en transport	16
3.2.4 Evolution en utilité	17
3.2.5 Evolution du nombre de liens par élément	17
3.3 Evolution de la représentation hiérarchique	18
3.4 Conclusion	18

4 Réduction du nombre de liens :	
 nouvel algorithme	20
4.1 Comment réduire le nombre de liens ?	20
4.2 La théorie	20
4.2.1 Un premier algorithme	20
4.2.2 Equivalence	21
4.2.3 Calcul d'erreur	22
4.3 Un nouveau raffinement : RaffinerEtRecolter	23
4.3.1 Limitation du nombre de liens créés	23
4.3.2 Utilisation des liens passifs	23
4.3.3 Critère de création des liens	24
4.3.4 Calcul récursif du facteur de forme	24
4.4 Nouvel algorithme	25
4.5 Résultats	26
4.6 Avantages et inconvénients	27
4.6.1 Avantages	28
4.6.2 Inconvénients	28
4.6.3 Comparaisons	29
4.7 Conclusion	29
5 Réduction de la hiérarchie	
 et émission retardée	30
5.1 Meilleure utilisation de la mémoire	30
5.1.1 Représentation de la radiosité	30
5.1.2 Etude de la complexité	31
5.1.3 Premiers résultats	31
5.2 Vers une hiérarchie réduite	32
5.3 Un nouvel algorithme : L'émission retardée	33
5.4 Conclusion	34
6 Conclusion	35
6.1 Etude des structures de radiosité hiérarchique	35
6.2 Nouvel algorithme	35
6.3 Réduction de la mémoire utilisée par la hiérarchie	36
6.4 Perspectives	36
Figures couleur	37
Bibliographie	40

Table des figures

1.1	Un trop grand nombre de liens	2
1.2	Récapitulatif des principaux sigles et responsables	3
2.1	Une petite explication des paramètres	6
2.2	Formulation matricielle de la radiosité avec subdivision [SP94]	9
2.3	Divers échanges[SP94]	10
2.4	Deux échanges à deux niveaux différents	10
2.5	Hiérarchie (Quad-tree)[SP94]	11
2.6	Push (a) Pull (b)[SP94]	11
3.1	Troupeau	14
3.2	Hall vide	14
3.3	Hall plein	14
3.4	Correspondance des hiérarchies	15
3.5	Evolution du nombre de liens au cours des itérations	16
3.6	Répartition de l'apport d'énergie sur les liens	16
3.7	Evolution des liens utiles (b) par rapport à ceux créés (a)	17
3.8	Evolution du nombre de liens par élément	18
3.9	Evolution du nombre de feuilles de la hiérarchie	18
4.1	Retour de raffinement d'un lien	24
4.2	Raffiner et Récolter	26
4.3	Bilan des liens stockés et comparaison avec le nombre de liens utiles	26
4.4	Nombre de liens stocké	27
4.5	Comparaison de la mémoire utilisée	27
4.6	Hall très éclairé	27
4.7	Mémoire utilisée sur nos scènes d'études	27
4.8	Temps de calcul pour 4 itérations	27
4.9	Nombre de liens recréés	28
5.1	Nouvelle organisation de la hiérarchie	31
5.2	Grand Hall	32
5.3	Test de l'émission retardée	34
6.1	Un trop grand nombre de liens	37
6.2	scènes d'étude	37
6.3	Grand Hall	38
6.4	Test de l'émission retardée	38

Chapitre 1

Introduction

1.1 Le sujet :

Illumination de scènes complexes par méthode de radiosité avec mémoire limitée
--

1.1.1 Motivation

De nos jours, l'imagerie prend une place de plus en plus importante dans tous nos moyens de communication, privilégiant ainsi la vue comme sens humain. Elle s'insère dans beaucoup de domaines (architecture, geste médical, cinéma, communications, ...) comme aide à la visualisation, à la conception, voire même comme outil de simulation (entraînement, mesure, visualisation, ...). Dans tous les cas, une des exigences primordiales est l'impression de réalisme (voir même la réalité) du rendu. En effet, la modélisation géométrique d'un objet n'est qu'un des aspects de sa représentation. De nombreux paramètres interviennent pour que nous ayons la sensation de son existence. Son interaction avec la lumière est un des principaux, puisqu'elle nous le rend visible. C'est pourquoi l'éclairage tient une place importante dans l'image de synthèse.

Dans ce domaine, la méthode de radiosité s'est imposée ces dernières années comme une des méthodes d'illumination globale offrant les solutions les plus réalistes. En effet, elle se base sur des équations de thermodynamique (échanges radiatifs) développées dans les années 1950, et, de plus contrairement à une méthode comme le lancer de rayon classique, elle considère l'éclairage de la scène 3D de façon globale, c'est à dire qu'elle considère tous les échanges lumineux possibles (éclairage direct des sources comme les réflexions secondaires). Cela permet donc une simulation des échanges énergétiques entre les divers objets d'une scène. De plus, le résultat obtenu est indépendant du point de vue, ce qui permet un affichage interactif de la scène.

Une des dernière évolution des algorithmes de radiosité, est la radiosité hiérarchique [HSA91], qui par une représentation multi-échelle des échanges et de la scène permet l'obtention d'un résultat plus rapide que les premiers algorithmes (résolution matricielle, radiosité progressive [CCWG88]). Cette structuration permet de tenir compte des différences d'importances des échange. Elle a été étendue à des groupes d'objets (clusters) [SAG94, Sil95] afin de rendre cet algorithme plus indépendant du nombre de polygones de modélisation.

Pendant, cette structuration est gourmande en place mémoire, notamment parce qu'elle doit stocker non seulement des liens entres les éléments, ces derniers indiquant un échange énergétique à un niveau donné (*figure 1.1*), mais aussi toute la hiérarchie des éléments (tous les niveaux de raffinement). Notre but sera donc de revoir cet algorithme afin de *réduire la mémoire* occupée pour pouvoir simuler des scènes de plus grande complexité (vers un million de polygones). En effet, celle-ci augmente plus vite que la puissance des machines à cause d'une exigence de réalisme de plus en plus grande. Ainsi, si par exemple,



FIG. 1.1: Un trop grand nombre de liens

une scène naturelle doit être simulée, la taille devient vite prohibitive par la modélisation en polygones (les feuilles, etc..).

1.1.2 Approche adoptée

Réduire la consommation de place mémoire d'un tel algorithme demande de bien comprendre sa structure et son fonctionnement. La radiosité ayant donné lieu à plusieurs algorithmes (avec des coûts en mémoire et en temps de calcul plus ou moins avantageux), il convient donc de faire une étude approfondie de ces différentes méthodes afin de déceler ce que pourrait être un algorithme de radiosité hiérarchique de coût mémoire moindre. Il est nécessaire de confronter aussi cette étude théorique à des études statistiques pour mieux cerner le comportement.

La radiosité hiérarchique stocke en mémoire à la fois la hiérarchie elle-même - les différents niveaux de représentation - mais aussi tous les liens établissant les échanges énergétiques. Pour chacune de ces deux structures, il faut donc, à la fois par une approche théorique, et par une approche statistique, voir où se fait une consommation « inutile » de mémoire.

Cela nous conduira à développer un nouvel algorithme, dont la validation se fera par comparaison des résultats avec ceux obtenues par l'algorithme existant, que ce soit pour la qualité du rendu ou pour la consommation de la mémoire.

1.2 Plan du rapport

Chapitre 2 : Etat de l'art

Ce chapitre a pour but de bien comprendre ce qu'est un algorithme de radiosité à travers les publications consacrées. Pour cela, nous allons nous attacher à bien définir non seulement les équations des échanges radiatifs, mais aussi le cadre des algorithmes de radiosité (discrétisation, restrictions). Bien que nous nous attarderons sur la compréhension de la radiosité hiérarchique, d'autres algorithmes seront introduits dans le but de pouvoir comprendre les avantages et les désavantages de celle-ci.

Chapitre 3 : Etude de la structure de radiosité hiérarchique

Afin de développer un nouvel algorithme réduisant le coût en place mémoire, il convient d'étudier d'une façon plus statistique le comportement de l'algorithme. Ces études vont nous permettre de visualiser les différents paramètres des échanges dans le but de mieux cerner comment aborder un nouvel algorithme. Notamment, nous présenterons des résultats sur le nombre de liens utilisés, le nombre de liens « utiles », afin de déterminer quels liens nécessite d'être stockés. L'évolution de la hiérarchie au cours des itérations sera aussi étudiée pour situer les moments de fort accroissement de la structure qu'il faudrait éviter.

Chapitre 4 Nouvel algorithme (Réduction du nombre de liens)

Tirant partie des informations des chapitres précédents, un nouvel algorithme est développé sur la

base d'un nouveau paramètre (la radiosité à transmettre), de l'utilisation de la hiérarchie des liens [Sha97, DS97] et d'un critère de création de liens. Cette algorithmes a pour avantages principaux, non seulement la réduction du nombre de liens utilisés, mais aussi une meilleure approximation des échanges, tout en conservant un temps de calcul raisonnable. Les résultats de notre implémentation seront aussi présentés.

Chapitre 5 : Réduction de la hiérarchie et émission retardée

La hiérarchie elle-même est grande consommatrice de place mémoire. Nous présenterons donc ici des perspectives quant à la réduction de cette consommation à la fois par une meilleure hiérarchisation, mais aussi par la l'utilisation d'une hiérarchie réduite. Nous décrirons aussi un autre algorithme développé mais non complètement testé, retardant l'émission d'énergie de certains éléments.

Chapitre 6 : Conclusion

Ce chapitre va nous permettre de faire un bilan de ce projet. Nous y ferons une synthèse des travaux effectués, des conclusions que nous en tirons, et des nouveaux objectifs qu'elles impliquent.

1.3 Présentation du cadre du projet : Le projet *i*MAGIS

Le projet se déroule au sein de l'équipe *i*MAGIS¹, encadré par Georges Drettakis², membre permanent. *i*MAGIS est un projet commun CNRS³ -INRIA⁴ -INPG⁵ -UJF⁶ Il fait partie du laboratoire GRAVIR-IMAG. L'équipe est actuellement composée de six chercheurs permanents et accueille une dizaine de doctorants.

	Signification	Responsable
GRAVIR	GRA phique, VI sion, RO botique	Claude PUECH
<i>i</i>MAGIS	MO dèles, AL gorithmes, GÉ ométrie pour le graphique et l' IM age de SY nthèse	Claude PUECH

FIG. 1.2: Récapitulatif des principaux sigles et responsables

Le projet *i*MAGIS se place dans le domaine de l'informatique graphique et de la synthèse d'images. Il a pour but le développement d'outils permettant de concevoir, puis d'utiliser dans le cadre d'applications de taille significative, et en particulier en vue de simulations, des maquettes numériques 3D. Ces maquettes peuvent être purement géométriques ou posséder également des propriétés «physiques»(photométriques ou mécaniques par exemple).

Les applications visées se situent dans des domaines très divers (construction automobile ou aéronautique, urbanisme, éclairagisme, bâtiment, téléphonie mobile, chimie, chirurgie assistée, agronomie, environnement, audiovisuel, etc...). Dans bien des cas il s'agit de concevoir les techniques (modélisation et algorithmes graphiques) sur lesquelles reposent les systèmes «de réalité virtuelle» (ou de «réalité augmentée») qui commencent à voir le jour. Le défi à relever est de leur fournir la puissance nécessaire à l'affichage et à l'interaction «temps réel» qui les caractérisent.

Les sujets de recherches en cours sont :

1. Visualisation d'environnements complexes.
2. Rendu réaliste, simulation de l'éclairage
3. Animation, modélisation d'objets déformables et de leur comportement
4. Algorithmique de la visibilité, structures de données efficaces pour le rendu de scènes très complexes

¹*i*MAGIS / GRAVIR-IMAG - BP 53 - 38041 Grenoble Cedex 9 - France

²Email : Georges.Drettakis@imag.fr

³Centre National de la Recherche Scientifique

⁴Institut National de Recherche en Informatique et en Automatique

⁵Institut National Polytechnique de Grenoble

⁶Université Joseph Fourier

5. Interactivité, réalité augmentée

Certains projets sont menés en partenariat avec des entreprises comme Renault (Simulation de conduite en milieu urbain), HP, ou avec d'autres centres de recherche comme le CIRAD⁷ (Simulation de photosynthèse et croissance de plantes), d'autres sont menés dans le cadre d'un partenariat européen (par exemple SIMULGEN et ARCADE, contrats européennes concernant la simulation de l'éclairage).

Pour la simulation d'éclairages, un logiciel implémentant l'algorithme de radiosit  hi rarchique avec clustering a  t  d velopp  : **Bright** (pour **R**ealistic **I**llumination with **G**lobal **H**ierarchical **T**ransfer, **B** d signant en g n ral la radiosit  dans les algorithmes). Il se d cline en plusieurs versions suivant l'orientation des diverses recherches. Celle qui est d velopp e au cours de ce projet est **RMbright** (RM pour Reduce Memory). Notre projet s'int gre donc dans le cadre des recherches sur le rendu et la visualisation d'environnements complexes du projet *iMAGIS*.

⁷Centre International de Recherche en Agronomie et D veloppement

Chapitre 2

Etat de l'art

Une grande partie de cette présentation de la radiosité à été faite à l'aide du livre de François Sillion et Claude Puech (Radiosity & Global Illumination[SP94]) et du rapport technique de Willmott et Heckbert (An Empirical Comparison of Radiosity Algorithms[WH97a, WH97b]) faisant une étude comparé des divers algorithmes de radiosité.

Inscrites dans le cadre de l'illumination globale, les méthodes de radiosité sont basées sur des équations-bilan d'échanges radiatifs qu'il convient donc d'introduire dans un premier temps. Par la suite, nous nous attarderons sur leur discrétisation en éléments finis, et aux différentes méthodes de résolutions, les méthodes de radiosité. Celles-ci seront présentées, dans le cadre de notre étude, avec un intérêt particulier pour leur consommation en place mémoire.

L'algorithme de radiosité hiérarchique, que nous voulons modifier, sera alors étudié plus en détail pour bien cerner sa structure et son comportement.

2.1 Introduction : Apport de l'illumination globale

Dans l'informatique graphique, l'impression de réalisme est depuis longtemps un des aspects essentiel dans le rendu des images. La première approche, locale, utilise comme seule variable l'intensité que fournit une source de lumière à une surface de couleur donnée (considéré comme propriété physique). Seul l'éclairage direct des objets est alors calculé. Pour un rendu plus réaliste, la solution généralement employée est le suivi du cheminement d'un rayon lumineux pour prendre en compte l'influence d'autres réflexions. Mais ce suivi peut ne pas considérer certaines réflexions.

Pour un rendu plus réaliste, il faut aussi tenir compte de la participation de tous les objets de la scène (c'est à dire toutes les réflexions indirectes entre objets). Les méthodes d'illumination globale simule les échanges de la scène d'un point de vue global, considérant toutes les réflexions secondaires. Elles se base sur les propriétés de réflexion des matériaux. La couleur d'un objet devient alors aussi une variable, conséquence de son interaction avec le milieu. Il y a pour cela deux approches, qui utilisant les différentes natures de la lumière, particulaire ou ondulatoire.

Les méthodes de Monte-Carlo considèrent la lumière comme une particule. Le rendu d'une scène se fait par le suivi de chemins aléatoires de rayons (la loi de probabilité correspondant aux propriétés de réflexion des objets). Le flux de photon convergeant vers l'image est alors simulé. Mais une telle méthode est dépendante du point de vue.

Les algorithmes de radiosité, eux, considèrent la lumière comme une onde dans un état stationnaire. Ils sont en fait des algorithmes de résolution par éléments finis d'équations thermodynamiques (flux d'énergie). Ils offrent ainsi des solutions indépendantes du point de vue, et avec un calcul d'erreur possible, permettant ainsi des simulations.

2.2 Présentation générale de la radiosité

2.2.1 Les échanges radiatifs

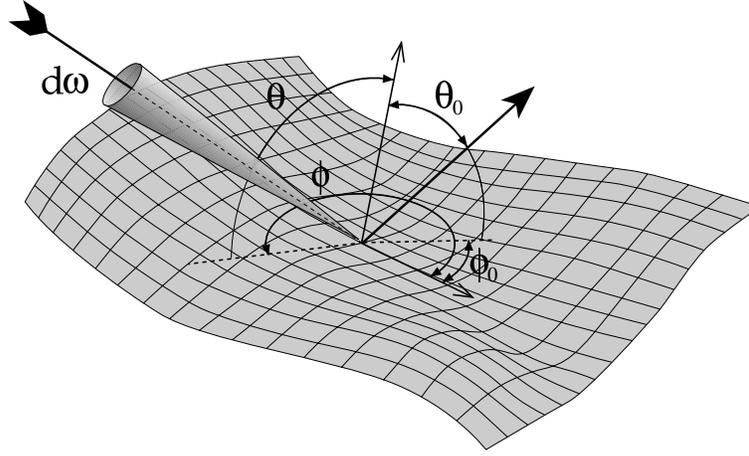


FIG. 2.1: Une petite explication des paramètres

Pour chaque longueur d'onde d'un spectre lumineux, seul l'état stationnaire (rapidement atteint pour des échanges à la vitesse de la lumière) est considéré. De plus, chaque objet de la scène possède pour propriété physique, outre une **radiance** (énergie qu'il transmet à la scène), une fonction de **reflectance** (BRDF pour Bidirectionnal Reflectance Distribution Function) qui indique en un point x d'une surface de la scène, la répartition de l'énergie renvoyée dans la direction de réflexion (θ_0, ϕ_0) en fonction de la direction d'incidence (θ, ϕ) (figure 2.1) :

$$\rho_{bd}(x, \theta_0, \phi_0, \theta, \phi)$$

En notant alors $L(x, \theta_0, \phi_0)$ l'énergie (**radiance** en $W.m^{-2}.sr^{-1}$) émise d'un point de la scène x dans la direction (θ_0, ϕ_0) (L_e pour l'émission propre de lumière et L_i pour la lumière incidente), l'équation-bilan des échanges au point x et dans la direction (θ_0, ϕ_0) s'écrit :

$$L(x, \theta_0, \phi_0) = L_e(x, \theta_0, \phi_0) + \int_{\Omega} \rho_{bd}(x, \theta_0, \phi_0, \theta, \phi) L_i(x, \theta, \phi) \cos\theta d\omega \quad (2.1)$$

($d\omega$ représente un angle solide et Ω l'espace des directions possibles en x).

En fait, ce formalisme représente les échanges pour une longueur d'onde fixée. Il faudrait donc intégrer cette égalité sur tout le spectre simulé pour obtenir le bilan global. Dans le domaine de la lumière visible, une représentation en longueurs Rouge-Vert-Bleu (RGB) est une approximation suffisante de l'ensemble des longueurs d'onde perçues. C'est cette approche qui est généralement employée.

2.2.2 Les équations de radiosité en milieu diffus

La radiosité représente en fait le flux d'énergie émise dans toutes les directions en un point de la scène (donc en $W.m^{-2}$). Elle est en général notée B . Elle est exprimée par :

$$B(x) = \int_{\Omega} L(x, \theta, \phi) \cos\theta d\omega \quad (2.2)$$

Dans le cas d'une réflexion diffuse idéale (c'est à dire indépendante de la direction), la reflectance et la radiance ne dépendent plus que de la position :

$$\rho_{bd}(x, \theta_0, \phi_0, \theta, \phi) = \frac{\rho(x)}{\pi} \quad (2.3)$$

et $L(x, \theta_0, \phi_0, \theta, \phi) = L(x)$, soit

$$B(x) = \frac{L(x)}{\pi} \quad (2.4)$$

Ainsi, en combinant les équations 2.1, 2.2, 2.3 et 2.4, on obtient :

$$B(x) = E(x) + \rho_d(x)I(x) \quad (2.5)$$

avec

$$E(x) = \int_{\Omega} L_e(x, \theta, \phi) \cos\theta d\omega = \frac{L_e(x)}{\pi}$$

comme **exitance** (radiosité intrinsèque au point x , pour les sources de lumière),

$$\rho_d(x) = \rho(x)/\pi$$

comme réflectance diffuse, et enfin

$$I(x) = \int_{\Omega} \frac{L_i(x, \theta, \phi)}{\pi^2} \cos\theta d\omega$$

comme **irradiance** (flux d'énergie reçue en x). Si y est un point de la scène est visible par x dans la direction (θ, ϕ) , alors $L_i(x, \theta, \phi) = L(y, \theta', \phi')$, (θ', ϕ') représentant l'angle sous lequel y voit x . Dans le cadre de réflecteurs diffus, $L(y, \theta', \phi') = L(y) = \pi B(y)$. Ainsi, en notant $V(x, y)$ la fonction qui indique la visibilité du point y par le point x ($V(x, y) = 1$ si x est visible de y , 0 sinon), le terme d'irradiance s'écrit alors :

$$I(x) = \int_S B(y) V(x, y) \frac{\cos\theta}{\pi} d\omega$$

(S est l'ensemble de la scène).

La formulation de l'équation de radiosité dans le cas diffus idéal est ainsi :

$$B = E + \mathfrak{R}(B) \quad (2.6)$$

où \mathfrak{R} est un opérateur linéaire sur la fonction de radiosité modélisant les échanges. La solution est donc le point fixe de cette équation.

2.3 Discrétisation

2.3.1 Formulation générale

Pour pouvoir résoudre l'équation 2.6 dans un cas plus général que celui qu'offrirait une méthode analytique, une méthode d'éléments finis est employée. Pour cela, la représentation de la scène est discrétisée. Pour cela, la radiosité et la réflectance sont supposées constantes sur chaque élément de discrétisation. Dans ce cas, en notant B_i la valeur moyenne de radiosité sur l'élément P_i d'aire A_i ,

$$B_i = \frac{1}{A_i} \int_{x \in P_i} B(x) dx$$

de même E_i est l'exitance moyenne de l'élément P_i , et ρ_i , sa réflectance.

L'équation 2.6 devient alors :

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j \frac{1}{A_i} \int_{x \in P_i} \int_{y \in P_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} V(x, y) dy dx \quad (2.7)$$

avec

$$d\omega = \frac{\cos\theta_j}{r^2} dy$$

soit

$$B_i = E_i + \rho_i \sum_{j=1}^N F_{i,j} B_j \quad (2.8)$$

où

$$F_{i,j} = \frac{1}{A_i} \int_{x \in P_i} \int_{y \in P_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} V(x,y) dy dx \quad (2.9)$$

est un terme purement géométrique appelé le **facteur de forme**. Il a pour propriété que $A_i F_{i,j} = A_j F_{j,i}$, mais surtout, comme il exprime la proportion reçue du flux d'énergie de l'émetteur (irradiance reçue sur la radiosité de l'émetteur) :

$$\sum_{i=1}^N F_{i,j} \leq 1 \quad (2.10)$$

2.3.2 Résolution Matricielle

Soit \mathbf{M} la matrice où $M_{i,j} = \delta_{i,j} - \rho_i F_{i,j}$. L'équation 2.8 devient alors

$$\mathbf{M}\mathbf{B} = \mathbf{E}$$

Du fait de la relation 2.10, M est à diagonale dominante (strictement ou non, suivant la réflectance moyenne de la scène). La méthode de Gauss-Seidel est donc justifiée, mais la résolution réagit bien aussi aux algorithmes de relaxation et au gradient conjugué [GTGB84].

Même si elle s'exprime facilement, cette méthode offre de nombreux désavantages [WH97b], notamment en place mémoire (elle occupe $O(N^2)$ si N est le nombre d'éléments de la scène) et en temps (le temps passé à calculer les facteurs de forme et la visibilité est assez prohibitif). De plus, la taille occupée étant proportionnelle au carré du nombre d'éléments de discrétisation, une subdivision en 4 de chacun pour un maillage plus fin augmenterait la taille par 16 !

Cependant, une autre formulation permet entre autre d'avoir un raffinement fonction des variations de valeurs de radiosité sur les objets tout en conservant une plus petite quantité d'information [CGIB86]. Les échanges sont considérés à deux niveaux :

1. Solutions des échanges entre les objets eux-mêmes (*figure 2.2(3)*).
2. Solution des échanges entre les objets et les éléments (subdivision d'un objet) de la scène (*figure 2.2(4)*).

Dans tous les cas, là aussi la convergence est fortement dépendante de la réflectance maximale de la scène, car elle influence la dominance diagonale de la matrice.

Elle reste tout du moins pratique pour les petites scènes.

2.3.3 La radiosité progressive

Pour éviter le coût de stockage en $O(N^2)$, l'algorithme de radiosité progressive [CCWG88] ne considère qu'un seul émetteur, le plus fort, 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. Il a été

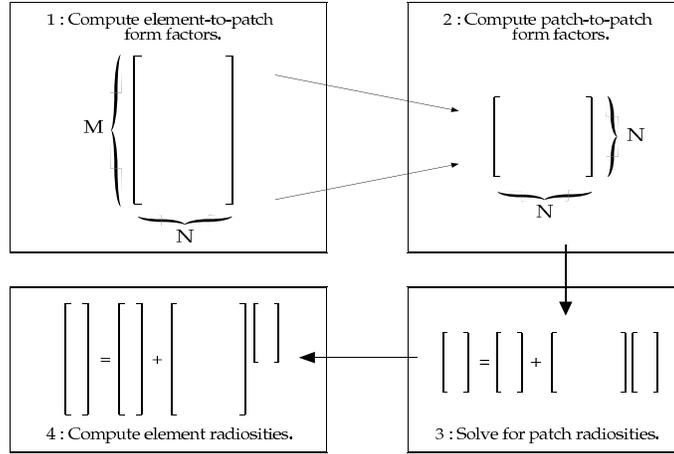


FIG. 2.2: Formulation matricielle de la radiosité avec subdivision [SP94]

démontré [WH97b] que la convergence de ces algorithmes se fait en $O(N^2)$.

$S = E$ // radiosité non-transmise de chaque élément (shooting)
 $B = 0$ // radiosité totale de chaque élément **Tant que** $\|B\| \geq \text{erreur}$ **faire**
 soit j l'indice du $\text{Max}(S)$: alors $\Delta\text{Rad} = S[j]$ et $S[j] = 0$.
pour tous les éléments e de la scène **faire**

$$\begin{cases} B[e] = B[e] + \rho_e F_{e,j} V(e, j) \Delta\text{Rad} \\ S[e] = S[e] + \rho_e F_{e,j} V(e, j) \Delta\text{Rad} \end{cases}$$

Avec une telle formulation, la matrice des facteurs de forme n'a plus besoin d'être stocké, car une seule ligne peut être utilisée, celle-ci devant être recalculée pour chaque itération. Le coût de stockage devient alors en $O(N)$, mais avec des baisses de performances au point de vue temps de calcul (occupé par le calcul des facteurs de forme). De plus, cette approche permet un raffinement adaptatif suivant le gradient de radiosité, pour obtenir une meilleure représentation des détails (ombres, ...).

$S = E$ // radiosité non-transmise de chaque élément (shooting)
 $B = 0$ // radiosité totale de chaque élément **Tant que** $\|B\| \geq \text{erreur}$ **faire**
 soit j l'indice du $\text{Max}(S)$: alors $\Delta\text{Rad} = S[j]$ et $S[j] = 0$
pour tous les éléments e de la scène **faire**

$$\begin{cases} B[e] = B[e] + \rho_e F_{e,j} V(e, j) \Delta\text{Rad} \\ S[\text{pere}(e)] = S[\text{pere}(e)] + \frac{A_e}{A_{\text{pere}(e)}} \rho_e F_{e,j} V(e, j) \Delta\text{Rad} \end{cases}$$

pour tous les éléments e de la scène **faire si** $\text{gradient}(e) > \varepsilon$

$$\begin{cases} B[e] = B[e] - \rho_e F_{e,j} V(e, j) \Delta\text{Rad} \\ \text{subdivise}(e) \\ \forall c \in \text{fils}(e) B[c] = B[c] + \rho_c F_{c,j} V(c, j) \Delta\text{Rad} \end{cases}$$

Ceci, tant qu'il y a eut des subdivisions

Un autre gros désavantage de cette méthode est l'absence d'un contrôle d'erreur et de la convergence, 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 corriger cette imprécision.

2.4 La radiosité hiérarchique

A travers les méthodes précédentes, l'intérêt du choix d'une stratégie de raffinement s'est affirmé pour une bonne représentation de la fonction de radiosité. Certains lieux de la scène demande une représentation plus fine pour une meilleure approximation de la fonction de radiosité. Un maillage adaptatif est apparu avec la radiosité progressive (*voir ci-dessus*), mais cette méthode reste trop rigide (seuls les objets peuvent être émetteurs et il n'y a pas de maîtrise de l'éclairage secondaire). Une représentation multi-résolution des échanges permet, elle, d'apporter en plus du maillage adaptatif, une maîtrise de l'erreur global par une maîtrise de l'erreur locale.

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

2.4.1 Principes

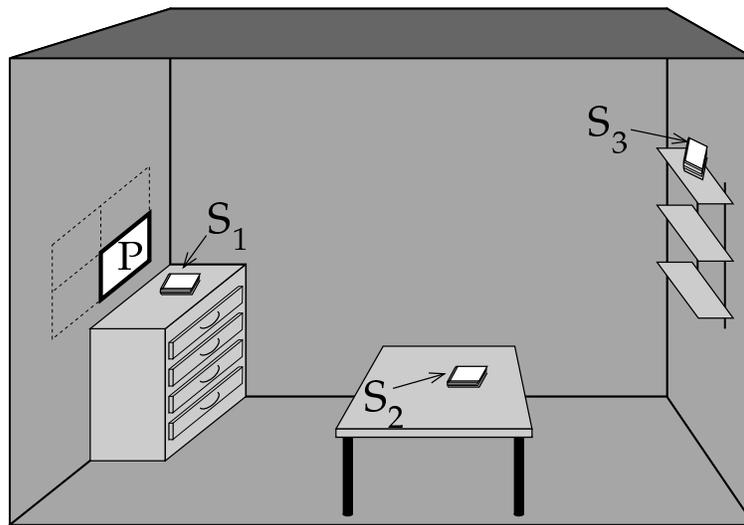


FIG. 2.3: Divers échanges[SP94]

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. Par exemple, en regardant la figure 2.3, le livre S_1 influencera beaucoup plus la surface P que le livre S_3 . De même, une représentation au niveau P du mur dans ce dernier échange peut être trop fine pour un niveau d'erreur donné. La radiosité hiérarchique tient compte de ces différences et ne simule un échange qu'au «juste» niveau de hiérarchie. C'est une représentation multi-résolution des échanges et de la radiosité. Les feuilles de la hiérarchie sont la base de discrétisation de la fonction de radiosité (*figure 2.5*).

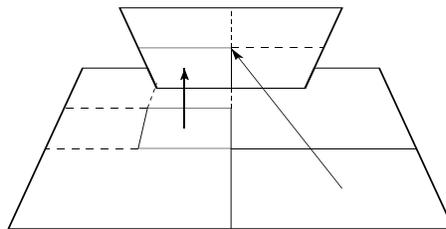


FIG. 2.4: Deux échanges à deux niveaux différents

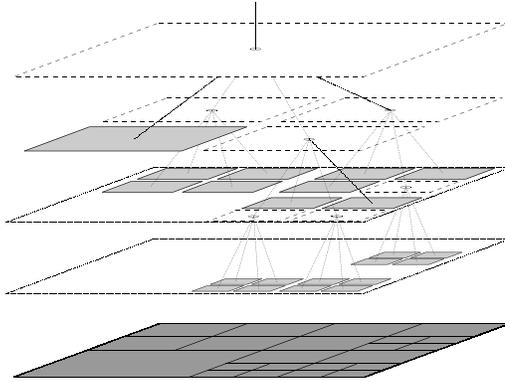


FIG. 2.5: Hiérarchie (Quad-tree)[SP94]

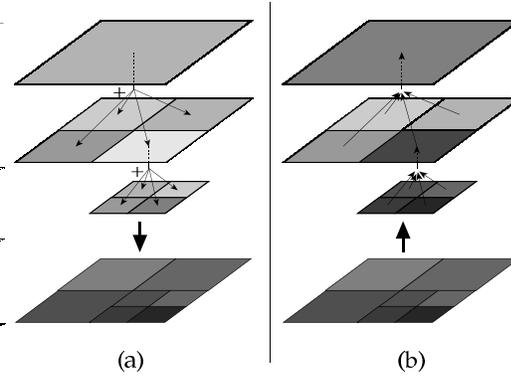


FIG. 2.6: Push (a) Pull (b)[SP94]

Des liens entre les éléments (conservant des informations comme le facteur de forme pour ne pas avoir à le recalculer) permettent d'établir les échanges énergétiques. Ces liens modélisent l'interface entre un émetteur et un récepteur. Ils sont stockés sur les récepteurs pour retrouver les émetteurs (figure 2.4). Cette représentation permet un raffinement dynamique. Le calcul de radiosité, pour chaque itération se fait en trois passes :

1. Raffinement (*refine()*) : 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 transportée ($F_{i,j}B_j$), l'énergie ($A_iF_{i,j}B_j$), et bien d'autres.
2. *gather()* («récolte») : chaque élément calcule son irradiance grâce aux liens
3. *pushpull()* (figure 2.6) : un parcours récursif de la hiérarchie permet de descendre l'irradiance de chaque élément à ses fils (ajout de son irradiance à celle de ses fils) et en remontant la radiosité (moyenne des radiosités des fils pondérée par les aires). La mise à jour de la radiosité se fait alors sur les feuilles de la hiérarchie.

C'est une résolution multi-résolution (ondelettes) de l'équation 2.8.

2.4.2 Notations

Dans tout ce chapitre et dans ceux qui suivent, nous allons garder les mêmes notations. Il convient donc de les fixer un fois pour toute.

Les notations suivantes sont définies :

- \mathbb{E} : Ensemble des éléments de la hiérarchie (représentation multi-résolution des objets de la scène)
- \mathbb{S} ($\mathbb{S} \subset \mathbb{E}$) : Ensemble des sources (éléments d'exitance non nulle). Elle représente un vecteur de radiosité \mathbf{E}
- \mathbb{L}_e ($\forall e \in \mathbb{E}$) : Ensemble des éléments liés à l'élément e (e est considéré comme un récepteur)
- I_e ($\forall e \in \mathbb{E}$) : Irradiance de l'élément e
- B_e ($\forall e \in \mathbb{E}$) : Radiosité de l'élément e
- ρ_e ($\forall e \in \mathbb{E}$) : Réflectance de l'élément e . Elle est aussi noté de façon matricielle par $\boldsymbol{\rho}$.
- \mathbf{F} : matrice des facteurs de forme.

Il faut noter que, pour ces 3 dernières valeurs, nous ne travaillons que pour une longueur d'onde fixée (paragraphe 2.1.2). Il faut donc refaire les mêmes calculs pour chaque longueur d'onde de la représentation (par exemple, 3 pour une représentation RGB).

2.4.3 Algorithme

L'algorithme hiérarchique se base sur deux variables pour chaque élément de la hiérarchie : I_e et B_e . Toute la radiosit  est recalcul e   chaque it ration ($\mathbf{B}^{i+1} = \mathbf{E} + \rho\mathbf{F}\mathbf{B}^i$) pour chaque  l ment ce qui limite l'erreur totale faite. Cela correspond   une recherche de point fixe.

Avant la premi re it ration, seules les sources de lumi re poss dent une radiosit  non-nulle. ($\forall e \in \mathbb{E}, B_e^0 \neq 0 \iff e \in \mathbb{S}$). Afin de garder la coh rence de la repr sentation multi- chelle, un Pull() ( quation 2.14) est n cessaire pour la mettre   jour.

Cette pr cision  tant faite, voici l'algorithme pour l' tape i :

1. *gather()* : L'irradiance re ue sur chaque  l ment de la hi rarchie (e) est calcul e, gr ce aux liens.

$$I_e = \sum_{s \in \mathbb{L}_e} F_{e,s} B_s^i \quad (2.11)$$

2. *push()* : L'irradiance re ue par chaque feuille de la hi rarchie (son irradiance plus celle de tous les noeuds dont elle d rive) est calcul e, soit, en descendant la hi rarchie,

$$\forall e \in \mathbb{E}, I_{fus(e)} = I_{fus(e)} + I_e \quad (2.12)$$

3. Sur les feuilles l de la hi rarchie : Mise   jour de la radiosit  de feuilles.

$$B_l^{i+1} = \rho_l I_l \quad (2.13)$$

4. *pull()* : Mise   jour de la radiosit  de chaque  l ment e , ceci en remontant la hi rarchie.

$$B_e^{i+1} = \frac{\sum_{c \in fus(e)} A_c B_c^{i+1}}{\sum_{c \in fus(e)} A_c} \quad (2.14)$$

La complexit  de cet algorithme en $O(k^2 + n)$, k  tant le nombre initial d' l ments, n  tant le nombre d' l ments apr s subdivision.

2.4.4 Stabilit 

Etudions la stabilit  de cette m thode, en consid rant que les erreurs proviennent des approximations faites sur les  changes. Elles seront repr sent es par un vecteur ϵ^{i+1} . En surmontant d'un \sim les valeurs entach es d'erreurs, la radiosit  calcul e   l' tape $i + 1$ s' crit :

$$\tilde{\mathbf{B}}^{i+1} = \mathbf{E} + \rho\mathbf{F}\tilde{\mathbf{B}}^i + \epsilon^{i+1}$$

soit, par r currence

$$\tilde{\mathbf{B}}^{i+1} = \left(\sum_{j=0}^{i+1} (\rho\mathbf{F})^j \right) \mathbf{E} + \sum_{j=0}^{i+1} (\rho\mathbf{F})^{i+1-j} (\epsilon^j)$$

$$\tilde{\mathbf{B}}^{i+1} = \mathbf{B}^{i+1} + \sum_{j=0}^{i+1} (\rho\mathbf{F})^{i+1-j} (\epsilon^j)$$

donc

$$\|\tilde{\mathbf{B}}^{i+1} - \mathbf{B}^{i+1}\| \leq \frac{\epsilon}{1 - \|\rho\mathbf{F}\|} \quad (2.15)$$

Cet algorithme voit donc aussi sa stabilit  d pendre largement de la r flectance maximum de la sc ne.

2.4.5 Les clusters

Pour simuler des scènes complexes, le complexité quadratique en nombre de polygones initiaux. Le clustering [SAG94, Sil95, CLSS97] propose une extension à la radiosité hiérarchique pour tenir compte des échanges à un niveau supérieur, c'est à dire en objets ou groupe d'objets. Alors que jusqu'alors nous ne considérons que des surfaces, nous allons supposer maintenant qu'à un certain niveau, un volume (ou cluster) est suffisant pour approcher le comportement des échanges (par exemple, dans la figure 2.2, nous ne considérerions que l'influence de l'étagère sur la surface P). La hiérarchie devient ainsi totale, la scène étant incluse dans un cluster racine. La complexité de l'algorithme devient en $O(N)$, N étant le nombre d'éléments initiaux de la scène.

En considérant que le comportement face aux échanges radiatifs d'un groupe de surfaces est isotrope, il peut être remplacé par une boîte englobante de comportement équivalent. Avec le tableau de correspondance suivant, les formules de calcul des facteurs de forme précédentes restent sous la même forme.

$$F_{i,j} = S_j S_i \frac{1}{\int_j dA_j} \int_i \int_j \frac{\tau \mathfrak{R}_i \Upsilon_q}{\pi r^2} dA_i dA_j$$

nom	symbole	surface	volume
Facteur du récepteur	\mathfrak{R}_i	$\cos\theta_i$	1
Facteur de l'émetteur	Υ_i	$\cos\theta_j$	1
Aire	A_i	A	$4\kappa V$
Echelle	S_p	1	$\frac{1}{4}$

(κ = coefficient d'extinction)

Le clustering apporte de nombreux avantages comme la réduction du nombre d'échanges, ou la simulation de milieux absorbants. Il a aussi été étendu aux comportements non-isotropes [SDS95, CLSS97].

2.4.6 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 [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 (facteur de forme F , BF , BFA , etc ..) ou sur un intervalle de confiance (Valeur maximal-Valeur minimale) [GH96], mais toujours avec l'objectif de réduire l'erreur d'approximation par la minimisation de ces valeurs.

Le raffinement se fait alors par un parcours de la hiérarchie afin, dans un premier temps, de trouver les liens sur chaque élément pour pouvoir, dans un second temps, les raffiner et ainsi les descendre dans la hiérarchie, car la valeur de radiosité étant croissante, le raffinement doit être de plus en plus fin.

2.4.7 Conclusion

La radiosité hiérarchique s'est imposée comme la méthode de radiosité la plus rapide à ce jour, ceci grâce à la hiérarchisation qu'elle fait des échanges. Cependant, même avec du clustering, la simulation de l'éclairage d'une scène de 40 000 polygones requière une place mémoire de 70 Mo. Afin d'accroître les possibilités d'un tel algorithme (comme augmenter la taille des scènes possibles pour la simulation), il convient de réduire ce coût.

Or, la radiosité hiérarchique est constituée de deux composantes principales qui sont, d'une part, la structure hiérarchique des éléments, et d'autre part, l'ensemble des liens. Il faut donc étudier l'évolution de l'utilisation de la mémoire de ces deux structures, mais aussi saisir plus précisément le comportement de la radiosité hiérarchique, afin de déterminer quelles seraient les modifications algorithmiques possibles pour réduire le coût théorique. Dans un premier temps, nous nous concentrerons sur l'étude des liens avant de travailler sur la hiérarchie elle-même.

Chapitre 3

Etude de la structure de radiosité hiérarchique

3.1 Introduction

Pour bien situer le problème à étudier, c'est-à-dire trouver en quoi l'algorithme de radiosité hiérarchique actuel utilise «trop de mémoire», nous allons étudier l'utilisation de la mémoire de ses deux structures principales, la hiérarchie elle-même et l'ensemble des liens qui semble être primordial[WH97a].

Des scènes ont donc été construites pour contenir différentes possibilités, comme de grandes surfaces qui seront donc beaucoup raffinées - création d'une forte hiérarchie -, de nombreuses sources (figures 3.2 et 3.3) - ce qui créera donc beaucoup de liens vers et à partir de celles-ci -, ou peu (figure 3.1), avec ou peu de clustering, ceci afin de voir l'influence de ces divers critères.

*Ces mêmes figures se trouvent en couleur
et avec une indication de maillage en fin de document : figure 6.2*



FIG. 3.1: Troupeau
43464 polygones
24338 clusters
1 source
raffiné en 71688 feuilles
de la hiérarchie



FIG. 3.2: Hall vide
94 polygones
1 cluster
16 sources
raffiné en 8962 feuilles
de la hiérarchie



FIG. 3.3: Hall plein
40886 polygones
22141 clusters
16 sources
raffiné en 57578 feuilles
de la hiérarchie

Pour chaque calcul, le critère utilisé compare le transfert d'énergie (BFA, c'est à dire Radiosité * Facteur de Forme * Aire du récepteur), à une borne fixée par l'utilisateur. Il a donc tendance à minimiser le transport d'énergie sur chaque lien.

3.2 Les liens

Dans le cadre de la radiosité hiérarchique classique, il semble impossible de réduire l'ensemble des liens, car chacun est essentiel à un échange. Cependant, nous allons essayer de déterminer en quoi cette structure pourrait consommer inutilement de la mémoire. Pour cela, trois angles d'étude de leur évolution ont été sélectionnés :

- le nombre : Cette étude va nous aider à déterminer quels sont les moments où la majorité des liens est créée, ceci afin d'éviter leur création dans un nouvel algorithme.
- les transports : Le but est ici de chercher à déterminer si certains liens n'ont qu'une influence minimale au cours d'une itération, et à trouver un autre critère de création.
- l'utilité : Ici, nous allons plutôt étudier l'évolution de l'ensemble de liens nécessaires aux échanges. Pour cela, un lien est considéré utile si, au cours d'une itération, le surplus d'énergie qu'il apporte (radiosité, irradiance, ...) vérifie le critère d'établissement, c'est à dire, qu'il est nécessaire à l'échange courant. Nous définissons ce surplus comme la différence entre la valeur calculée à l'itération précédente par rapport à celle calculée pour l'itération courante (par exemple, pour la radiosité, $B^i - B^{i-1}$). Cet ensemble représente le nombre maximal de liens à ne pas dépasser pour chaque itération.

Pour suivre l'évolution des divers paramètres de l'éclairage (radiosité, énergie, irradiance, ...), nous allons non pas considérer la valeur obtenue à une itération, mais la différence de valeur entre deux itérations.

Mais pour pouvoir faire cela, il nous faut utiliser une autre structure dans la radiosité hiérarchique, la hiérarchie des liens.

3.2.1 Utilisation des liens passifs

Le raffinement crée lui aussi une hiérarchie sur les liens[Sha97, DS97]. En effet, en posant qu'un lien a est fils du lien b si a a été obtenu en raffinant b (voir figure 3.4), une structure d'arbre peut être construite. Cette nouvelle hiérarchie a été en particulier introduite pour permettre l'application de la radiosité hiérarchique à des environnements dynamiques[Sha97, DS97], car cette structure permet de détecter rapidement les lieux de recalcul quand un objet se déplace dans la scène. Pour le *Gather()*, seuls comptent les liens qui ne sont pas raffinés (ils sont au «juste niveau» d'échange). Ils sont alors appelés les liens *actifs*. Par opposition, ceux qui ont été raffinés seront appelés les liens *passifs*.

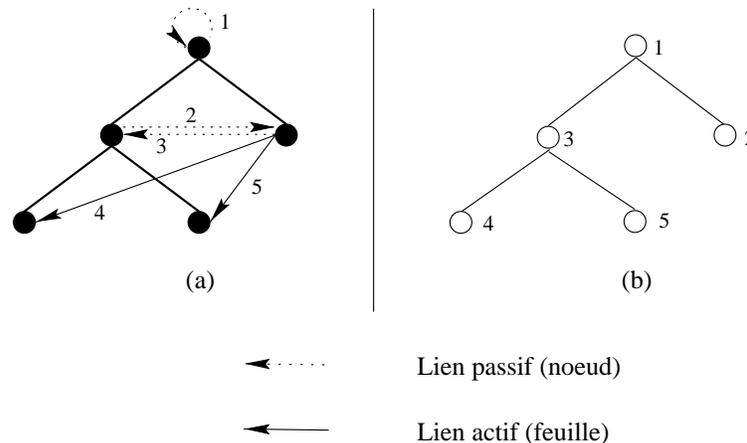


FIG. 3.4: Correspondance des hiérarchies
 (a) hiérarchie des éléments avec les liens, (b) hiérarchie des liens.
 Notons que 2 et 3 sont fils de 1, et que 4 et 5 sont fils de 3

Pour notre étude des liens, nous utiliserons cette nouvelle hiérarchie afin de pouvoir faire des tests sur tous les liens, actifs ou passifs. Comme nous conservons tous les liens du raffinement, cela permettra de déterminer, par exemple, si pour une itération, certains des liens que nous n'aurions pas conservé satisfait le critère de «juste niveau» (tous ses liens fils ne sont donc pas à conserver). Sans cette structure, il aurait fallu remonter l'historique de la création d'un lien à partir de celui-ci pour retrouver les liens pères.

3.2.2 Evolutions en nombre

A refaire !!!

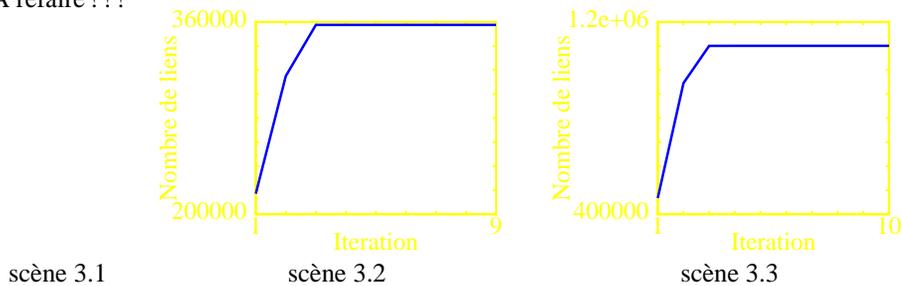


FIG. 3.5: Evolution du nombre de liens au cours des itérations

Avec l'algorithme classique [HSA91, Sil95], le nombre de liens ne peut qu'augmenter (figure 3.5). En effet, la radiosité de chaque élément est une fonction croissante du nombre d'itérations. Cette croissance a pour conséquence le raffinement de chaque lien (paragraphe 2.4.6). Ainsi, même pour les dernières itérations qui ne servent qu'à «corriger» faiblement les valeurs de radiosité, nous conservons cette lourde structure de liens, où chacun n'a qu'une très faible influence.

Ainsi, à la convergence de la simulation, ce nombre tend vers un maximum de liens, chacun étant situé au «juste niveau» de la somme des échanges (réflexions indirectes) au cours de toutes les itérations. Il faudrait donc pouvoir réduire ce maximum.

Ce que nous pouvons déjà remarquer, c'est qu'une grande partie des liens est créée à la première itération, c'est à dire en provenance des sources de lumière. Or la radiosité de celles-ci reste constante : c'est leur émittance. L'énergie une fois transmise par les liens créés à la première itération sera toujours identique dans les itérations suivantes. Le même calcul sera donc répété autant de fois. Ces liens devraient donc ne pas être stockés.

3.2.3 Evolution en transport

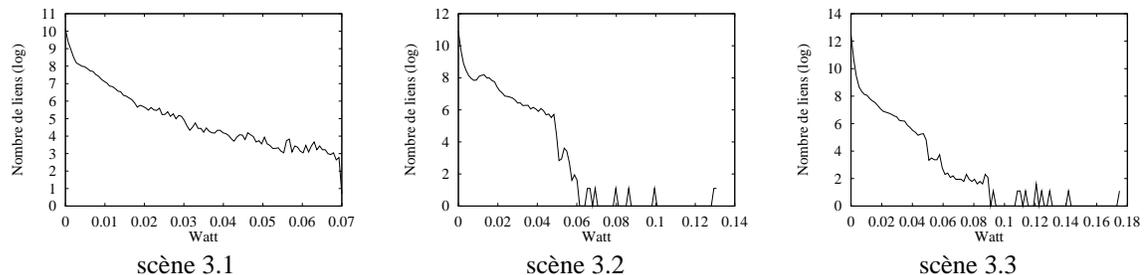


FIG. 3.6: Répartition de l'apport d'énergie sur les liens
 Cette valeur est celle que le critère BFA cherche à minimiser

Afin de déterminer en quoi un lien est utile au cours des itérations, l'étude va alors porter sur les différences d'énergie qu'il transporte (radiosité, irradiance et énergie) comme cela a été défini dans l'introduction de ce paragraphe.

Que nous considérons alors comme énergie transportée par les liens, la radiosité des émetteurs et des récepteurs, l'irradiance, ou l'énergie reçue (*figure 3.6*), du fait du critère de raffinement, la plupart en transporte une faible valeur. En effet, ces valeurs sont minimisées par les critères de raffinement, quelques soient ces derniers. Les figures 3.6 montrent cette répartition au cours d'une itération (la troisième) pour toutes les scènes d'étude. Ces courbes gardent une même allure sur toutes les itérations. La troisième a été choisie pour ses graphiques représentatifs.

Cette répartition nous interdit alors de ne pas considérer les liens de trop faible énergie transportée, car ils sont nombreux et leur suppression créerait alors un trop forte erreur. Par contre, il serait intéressant qu'un maximum de liens transportent une énergie de même amplitude, proche du critère de raffinement (moins de liens transportant plus d'énergie). Une subdivision non régulière (adaptée au gradient de la radiosité par exemple) permettrait de se rapprocher de ce but[HOL96].

3.2.4 Evolution en utilité

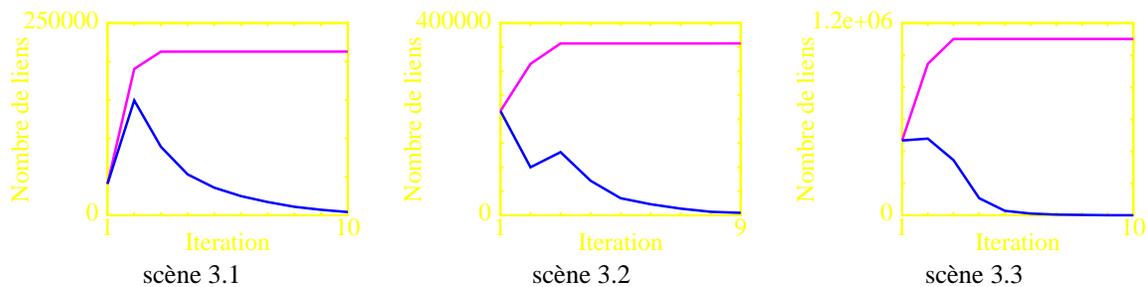


FIG. 3.7: Evolution des liens utiles (b) par rapport à ceux créés (a) par l'algorithme de radiosité hiérarchique

Dans la figure 3.7, nous présentons le nombre de liens utiles par rapport à celui de liens créés. Nous rappelons qu'un lien est dit utile, si le surplus d'énergie qu'il transporte (différence entre la valeur de l'itération courante et celle de l'itération précédente), comparé au critère de raffinement, fait qu'il n'a pas besoin d'être raffiné.

L'étude de cette évolution va nous permettre de voir le nombre maximal de liens que nous voudrions stocker à chaque itération. Comme nous pouvons le constater sur la figure 3.7, ce nombre a très vite tendance à diminuer. Il reste de plus très inférieur au nombre de liens actuellement créés par l'algorithme classique[HSA91, Sil95].

Ce qui serait intéressant, ce serait de ne stocker que le nombre de liens utiles à l'itération suivante. Un gain conséquent du nombre de liens (comme le montre la figure 3.7) pourrait donc être obtenue. Mais dans le cas d'un éclairage faible (scène 3.1), cela n'est pas évident, car le nombre de liens utiles augmente à la deuxième itération. Un stockage plus approprié serait de ne conserver que les liens qui seront soit utiles à l'itération suivante, soit nécessiteraient un raffinement.

3.2.5 Evolution du nombre de liens par élément

Dans la figure 3.8 nous présentons l'évolution du nombre de liens par élément. Ce que nous pouvons remarquer grâce à celle-ci, c'est que le nombre de liens par élément est très variable. Il dépend de la nature de la scène. Plus il y a de sources et d'éléments qui dépendent de la racine de la hiérarchie, plus ce nombre va être grand. De plus, les clusters ont tendance à réduire cette valeur, car, non seulement ils augmentent la taille de la hiérarchie, mais aussi, ils diminuent le nombre de liens nécessaires en considérant les échanges

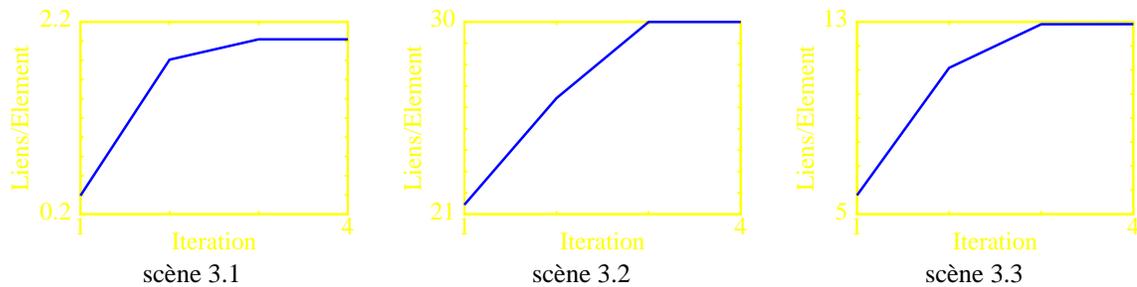


FIG. 3.8: Evolution du nombre de liens par élément

sur des groupes d'objets. Cependant, même si un lien est une structure beaucoup plus légère qu'un élément hiérarchique (rapport d'environ 1/10 dans Bright), l'ensemble des liens peut occuper une place aussi importante que la hiérarchie elle-même. C'est pourquoi il convient de le réduire, sans oublier que la structure hiérarchique elle-même est grande consommatrice de mémoire.

Ce que nous pouvons tout de même remarquer, c'est qu'une grande partie du nombre de liens par élément est, comme au paragraphe 3.2.2, due aux sources lumineuses.

En considérant alors le nombre de liens utiles par élément, il atteint un maximum inférieur à ce que nous aurions si tous les liens étaient conservés. Avec ce maximum, la hiérarchie elle-même peut devenir prépondérante dans l'occupation de la mémoire.

3.3 Evolution de la représentation hiérarchique

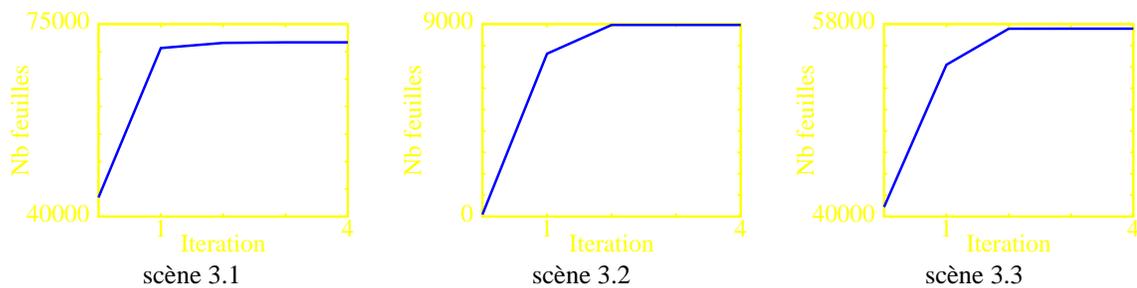


FIG. 3.9: Evolution du nombre de feuilles de la hiérarchie

La figure 3.9 nous permet de suivre l'évolution du maillage (nombre de feuilles de la hiérarchie) de la fonction de radiosit . Au cours des it rations, celui-ci s'adapte de plus en plus finement   la repr sentation de la radiosit . Cependant, nous pouvons remarquer qu'il est tr s vite stable. La majorit  du maillage se fait au cours des 3-4 premi res it rations.

Cela s'explique par le fait que l' nergie qui se propage est att nu e tr s vite non seulement gr ce aux facteurs de forme, mais aussi aux r flectances, ces deux param tres  tant chacun en g n ral inf rieur   un. Les it rations suivant cette stabilisation de la repr sentation hi rarchique servent alors seulement   atteindre l' tat d' quilibre des  changes lumineux.

3.4 Conclusion

Nous pouvons r sumer les conclusions de cette  tude ainsi :

1. Il paraît clair qu'en évitant le stockage des liens depuis les sources nous aurons un gain de place non-négligeable, puisque ceux-ci représentent une grande partie de l'ensemble des liens.
2. Le nombre de liens utiles est beaucoup plus faible, à chaque itération, que le nombre stocké par l'algorithme de la radiosité hiérarchique. Et cela ne cesse de s'accroître à chaque itération. Nous allons donc essayer de restreindre le stockage des liens au nombre réellement utile à chaque itération.
3. Vu le nombre de liens par élément et l'évolution de la taille de la hiérarchie des éléments, il conviendrait de réduire à la fois la hiérarchie et les liens.

Le fait le plus marquant, est que la grande majorité de la construction de ces deux structures se fait dans les 3-4 premières itérations, et se stabilise par la suite. Il faudrait donc orienter nos efforts sur celles-ci.

Dans un premier temps, nous allons développer un nouvel algorithme permettant de réduire le nombre de liens stockés. Dans un second temps, nous pourrions alors nous concentrer sur la hiérarchie et la complexité de représentation des scènes.

Chapitre 4

Réduction du nombre de liens : nouvel algorithme

4.1 Comment réduire le nombre de liens ?

Que nous apportent les liens dans l'algorithme de radiosit  hi rarchique classique ? Ils servent    tablir un  change   une pr cision donn e par le crit re de raffinement. Or, comme la valeur de radiosit  augmente sur chaque  l ment et que la valeur totale de radiosit  de chaque  l ment est utilis e pour le calcul, de plus en plus de liens seront n cessaires pour suivre la pr cision locale.

Si, comme dans la radiosit  progressive[CGIB86], nous ne transmettons que la radiosit  suppl mentaire qu'un  l ment re oit   chaque it ration, les sources, par exemple, n'interviendraient plus dans les it rations suivantes ( clairage indirect). Les liens qui deviendraient inutiles pourraient alors ne pas  tre stock s. Cette id e a aussi  t  d velopp e, simultan ment et ind pendamment   ce projet, au Computer Graphics Group de l'universit  de Erlangen-Nuremberg[MSSS98]. Une comparaison sera donc faite sur les choix effectu s et les r sultats (*paragraphes 4.5 et 4.6*).

4.2 La th orie

Dans l'algorithme classique de radiosit  hi rarchique(*cf paragraphe 2.4.3*),   chaque it ration, la radiosit  compl te de chaque  l ment est recalcul e. Cela est fait en deux passes, l'une r coltant l'irradiance sur chaque r cepteur   l'aide des liens (*Gather()* : * quation 2.11*), l'autre accumulant l'irradiance sur les feuilles de la hi rarchie et mettant   jour la repr sentation multi- chelle de la radiosit  (*PushPull()* : * quations 2.12 et 2.14*). Les liens sont donc conserv s au fil des it rations pour permettre ce recalcul total.

Notre nouvel algorithme, conservera la m me structure mais en se basant non plus sur la radiosit  totale, mais celle re ue   plus   chaque it ration. Comme celle-ci est d croissante apr s la premi re it ration, nous n'auront plus   stocker certains liens au m me niveau.

4.2.1 Un premier algorithme

Pour  viter le stockage de certains liens, nous stockons pour chaque  l ment de la hi rarchie la correction de radiosit    apporter. Pour toute la suite, nous allons l'appeler ΔB_e . Cela revient en fait   la s quence suivante sur les feuilles de la hi rarchie :

$$\begin{aligned} - \text{initialisation} : & \begin{cases} \mathbf{B}^0 = \mathbf{E} \\ \Delta \mathbf{B}^0 = \mathbf{E} \end{cases} \\ - \text{it ration } i + 1 : & \begin{cases} \Delta \mathbf{B}^{i+1} = \rho \mathbf{F} \Delta \mathbf{B}^i \\ \mathbf{B}^{i+1} = \mathbf{B}^i + \Delta \mathbf{B}^{i+1} \end{cases} \end{aligned}$$

La prise en compte de ce paramètre demande un nouvel algorithme assez proche de l'algorithme classique.

Avant la première itération, seules les sources de lumière possèdent une radiosité (propre et à transmettre) non-nulle.

$$\forall e \in \mathbb{E}, \begin{cases} B_e^0 \neq 0 \\ \Delta B_e \neq 0 \end{cases} \iff e \in \mathbb{S}$$

Rappelons que \mathbb{E} est l'ensemble des éléments de la hiérarchie, \mathbb{S} est l'ensemble des sources, et pour $e \in \mathbb{E}$, \mathbb{L}_e est l'ensemble des éléments liés à e . Il convient donc d'initialiser la représentation multi-résolution par un Pull() particulier, pour que sur chaque élément e , la relation $\Delta B_e = B_e$ (qui est déjà vérifiée sur les feuilles de la hiérarchie) soit conservée. En effet, à la première itération, la radiosité à transmettre est la radiosité totale. Pour cela il faut faire, en remontant la hiérarchie, et pour tous ses éléments e :

$$\begin{cases} \Delta B_e = \frac{\sum_{c \in \text{fils}(e)} A_c \Delta B_c}{\sum_{c \in \text{fils}(e)} A_c} \\ B_e^1 = \Delta B_e \end{cases} \quad (4.1)$$

Cela revient à considérer le Pull() (équation 4.5) avec $\mathbf{B}^{-1} = 0$. Après quoi, l'algorithme peut se dérouler et faire à chaque itération :

1. *gather()* : Pour tous les éléments e de la hiérarchie, faire

$$I_e = \sum_{s \in \mathbb{L}_e} F_{e,s} \Delta B_s \quad (4.2)$$

2. *push()* : en descendant la hiérarchie, et pour tous ses éléments e faire

$$\begin{cases} I_{\text{fils}(e)} = I_{\text{fils}(e)} + I_e \\ \Delta B_e = 0 \end{cases} \quad (4.3)$$

3. Sur les feuilles l de la hiérarchie :

$$\begin{cases} \Delta B_l = \rho_f I_l \\ B_l^{i+1} = B_l^i + \Delta B_l \end{cases} \quad (4.4)$$

La radiosité B_l^{i+1} sur les feuilles est donc la somme des radiosités reçues avant l'itération $i + 1$ et de la radiosité «à transmettre».

4. *pull()* : en remontant la hiérarchie, et pour tous ses éléments e faire

$$\begin{cases} \Delta B_e = \frac{\sum_{c \in \text{fils}(e)} A_c \Delta B_c}{\sum_{c \in \text{fils}(e)} A_c} \\ B_e^{i+1} = B_e^i + \Delta B_e \end{cases} \quad (4.5)$$

Après le PushPull(), la radiosité totale et la radiosité à transmettre ont toutes les deux une représentation multi-échelle.

4.2.2 Equivalence

Démontrons maintenant que cet algorithme est équivalent à l'algorithme classique, ceci pour un raffinement identique. Toutes les variables avec ^{mod} en exposant appartiennent au second algorithme et celles avec un ^{prec} indiquent les valeurs à la fin de l'itération précédente.

À l'état initial les valeurs de ΔB_e et de B_e^0 sont identiques dans les deux algorithmes sur les feuilles de la hiérarchie (seules les sources ont une radiosité non nulle et elle est égale à leur radiosité propre) et l'initialisation (équation 4.1) fait en sorte de garder cette propriété sur toute la hiérarchie \mathbb{E} . Tous les éléments e vérifient donc $\Delta B_e = B_e^0$. En considérant que $\forall e \in \mathbb{E}, B_e^{-1} = 0, B_e^0 = \Delta B_e + B_e^{-1}$ est vérifié. En effet, une partie de la démonstration repose sur le fait que $B_e^i = \Delta B_e + B_e^{i-1}$ à la fin de chaque itération.

Supposons qu'avant la i^{eme} ($i \geq 1$) itération cela soit vrai.

1. *gather()* : En utilisant l'équation 2.11 et le fait que l'on a à l'itération précédente $B_e^{i-1} = \Delta B_e^{prec,mod} + B_e^{i-2}$, celle-ci devient :

$$\forall e \in \mathbb{E}, I_e = \sum_{s \in L_e} F_{e,s} B_s^{i-2} + \sum_{s \in L_e} F_{e,s} \Delta B_s^{prec,mod}$$

soit, avec l'équation 4.2 et 2.11,

$$\forall e \in \mathbb{E}, I_e = I_e^{prec} + I_e^{mod} \quad (4.6)$$

2. *push()* : en descendant la hiérarchie,

$$\forall e \in \mathbb{E}, I_{fils(e)} = I_{fils(e)} + I_e$$

Soit $\forall e \in \mathbb{E}$

$$I_{fils(e)} = (I_{fils(e)} + I_e)^{prec} + (I_{fils(e)} + I_e)^{mod}$$

La relation $I_e = I_e^{prec} + I_e^{mod}$ est donc conservée dans la hiérarchie après le *push()*.

Sur les feuilles $B_l^i = \rho_f I_l$, soit $B_l^i = \rho_f I_l^{prec} + \rho_f I_l^{mod}$. Ainsi, les feuilles vérifient $B_l^i = B_l^{i-1} + \Delta B_l^{mod}$.

3. *pull()* : en remontant la hiérarchie,

$$\forall e \in \mathbb{E}, B_e^i = \frac{\sum_{c \in fils(e)} A_c B_c^i}{\sum_{c \in fils(e)} A_c}$$

donc

$$\forall e \in \mathbb{E}, B_e^i = \frac{\sum_{f \in fils(e)} A_f B_f^{i-1}}{\sum_{c \in fils(e)} A_c} + \frac{\sum_{f \in fils(e)} A_f \Delta B_f^{mod}}{\sum_{c \in fils(e)} A_c}$$

Soit $B_e^i = B_e^{i-1} + \Delta B_e^{mod}$

Ainsi, tous les éléments e de la hiérarchie vérifient $B_e^i = B_e^{i-1} + \Delta B_e^{mod}$. Or $B_e^{i-1} = B_e^{i-1,mod}$, donc $B_e^i = B_e^{i,mod}$. Les radiosités obtenues après l'étape i sont donc identiques du point de vue mathématique dans les deux algorithmes, pour un même raffinement, en ignorant les erreurs faites durant les calculs.

En effet, le problème de cette méthode est qu'elle rajoute une erreur à chaque itération, comme nous allons le voir dans le paragraphe suivant.

4.2.3 Calcul d'erreur

L'inconvénient de cet algorithme, c'est qu'il somme les erreurs de calcul de chaque itération. Il est donc plus instable que l'algorithme classique. En effet, si ϵ^{i+1} est le vecteur d'erreur de l'itération $i+1$, la valeur calculée est :

$$\tilde{\mathbf{B}}^{i+1} = \rho \mathbf{F} \tilde{\mathbf{B}}^i + \epsilon^{i+1}$$

Or comme $\mathbf{B}^{i+1} = \sum_{j=0}^{i+1} \Delta \mathbf{B}^j$, cela devient :

$$\mathbf{B}^{i+1} = \left(\sum_{j=0}^{i+1} (\rho \mathbf{F})^j \right) \mathbf{E} + \sum_{j=0}^{i+1} \sum_{l=0}^j (\rho \mathbf{F})^j \epsilon^{j-l}$$

Soit une borne d'erreur de

$$\|\tilde{\mathbf{B}}^{i+1} - \mathbf{B}^{i+1}\| \leq (i+1) \frac{\epsilon}{1 - \|\rho \mathbf{F}\|} \quad (4.7)$$

Le nombre d'itérations a donc tout intérêt à être limité (et le calcul de facteur de forme à être assez précis), alors que ce nombre n'influait pas l'erreur pour l'algorithme classique 2.15. Mais souvent, la radiosité hiérarchique converge en très peu d'itérations (aux alentours de 10). Cet algorithme reste alors généralement bon.

4.3 Un nouveau raffinement : RaffinerEtRecolter

Pour que le nouvel algorithme soit le plus efficace possible, il est nécessaire que le raffinement utilise dans son critère la radiosité à transmettre ΔB au lieu de la radiosité totale B . Comme nous l'avons constaté au chapitre précédent, les plus forts raffinements se font aux premières itérations. Or dans celles-ci, $\Delta B \approx B$. Donc le maillage reste presque identique à celui obtenu par un raffinement classique.

Alors que traditionnellement [HSA91, Sil95], le raffinement est dissocié du calcul des irradiances reçues (*Gather()*), il est ici nécessaire que les deux soient indissociables, ceci pour pouvoir calculer l'irradiance sans avoir à stocker un lien. Cela permettrait, en admettant l'anticipation du niveau auquel l'échange actuel se situerait dans les itérations suivantes, de savoir s'il doit être conservé (le niveau actuel restera bon ou sera raffiné) ou détruit (le niveau d'échange sera plus haut dans la hiérarchie). Dans tous les cas, il y a calcul de l'irradiance transmise (*Gather()*) pour que l'échange ait lieu, avec ou sans lien.

Cette approche nous conduira à utiliser une nouvelle structure, la hiérarchie des liens, ceci dans le but d'établir récursivement les liens. Cette structure apporte d'autres avantages (comme une amélioration de l'estimation des facteurs de forme) que nous détaillerons aussi.

4.3.1 Limitation du nombre de liens créés

Dans une première approche, il suffit de faire le calcul de l'irradiance transmise lorsqu'un lien est établi définitivement dans le raffinement pour une itération.

Cela fait, un critère de création des liens a été introduit afin d'anticiper le niveau d'échange futur. Nous avons défini trois cas :

1. Etablir le lien et calculer l'irradiance : cela correspond à un lien qui, soit est établi pour plusieurs itérations, soit doit être raffiné. C'est le critère par défaut.
2. Ne pas établir le lien mais calculer l'irradiance : cela correspond à un échange (lien) qui vérifie le critère de raffinement pour l'échange actuel, mais dont le «juste» niveau sera plus haut dans la hiérarchie à (aux) l'itération(s) suivante(s).
3. Ne faire aucun des deux : il n'y aura jamais aucun échange à ce niveau de la hiérarchie, ni à des niveaux plus fins.

Le cas «créer un lien mais ne pas calculer l'irradiance» ne signifie rien, car un lien est le lieu d'un échange.

Ces trois critères nous montrent la nécessité de faire un raffinement récursif, pour, par exemple, créer un lien à un niveau supérieur à celui où lieu l'échange actuel (*critère 2*). Un lien est alors établie dans 2 cas :

1. Sur les feuilles de la récursion : Il vérifie le critère de raffinement et le critère de création.
2. En retour de récursion : Il n'y a pas eu création du lien mais calcul d'irradiance (*critère 2*), et des liens ont été établis aux niveaux inférieurs. Ceci permet de garder la cohérence des échanges (un échange n'est pas répété sur plusieurs niveaux : *figure 4.1*)

4.3.2 Utilisation des liens passifs

Cette récursion ne peut pas être totale sans l'utilisation de la structure d'arbre sur les liens. En effet, à chaque nouvelle itération, le raffinement doit parcourir la hiérarchie pour trouver les liens de chaque élément et les raffiner si nécessaire. Chaque lien aurait donc perdu la notion de parenté (il est le produit du raffinement de quel lien ?). Le nombre de liens ne pourrait alors qu'augmenter .

La hiérarchie des liens (introduite au paragraphe 3.2.1) va donc être utilisée, pour pouvoir généraliser la récursion. Le raffinement se fera en parcourant non plus la hiérarchie des éléments, mais la hiérarchie des liens. Un lien actif peut toujours être raffiné, mais les liens passifs peuvent devenir actifs lorsque tous les critères sont vérifiés. Dans ce dernier cas, tous leurs fils doivent être éliminés de la hiérarchie (un lien actif en est forcément une feuille).

Il peut sembler surprenant que pour réduire le nombre de liens, on introduise des liens inutiles. Cependant, si α est le nombre moyen de fils dans la hiérarchie ($\alpha \geq 2$) et N le nombre de liens actifs, le surcoût

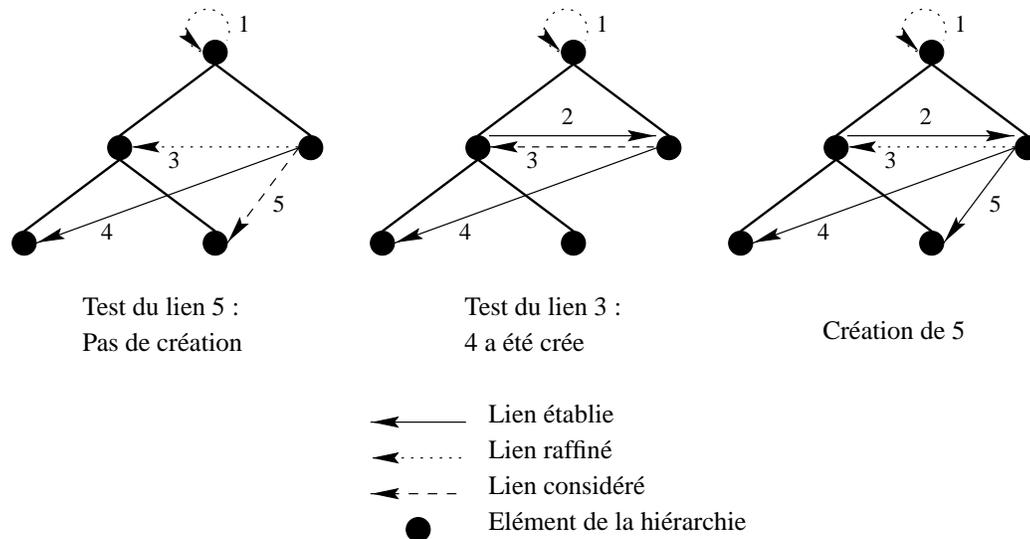


FIG. 4.1: Retour de raffinement d'un lien

est alors borné par $(\alpha - 1)^{-1}N$ (voir aussi [Sha97]). De plus, comme cette structuration a pour but de réduire le nombre de liens actifs, le nombre de liens passifs sera ainsi réduit au cours des itérations.

De plus, cette structure peut ne pas être explicitement stockée. En préservant sur un lien passif une information (il a été raffiné en subdivisant le récepteur ou l'émetteur), nous pouvons aller retrouver facilement les liens fils sur les fils du récepteur (ou de l'émetteur) courant. Il suffit alors d'un bit pour cette information. Dans notre implémentation, un lien passif occupe la même place qu'un lien actif.

4.3.3 Critère de création des liens

Nous devons utiliser un critère permettant l'évaluation de l'établissement des liens, ceci en bout de hiérarchie, et en retour de récursion si aucun lien n'a encore été créé. Son utilisation sera plus détaillée dans l'explication de l'algorithme (paragraphe 4.4).

Notre premier critère est de savoir si l'objet transmettant (ou recevant) de l'énergie est une source lumineuse pure. En effet, dans le cas récepteur, comme la radiosité est fixe, il est absolument inutile de faire un échange dans cette direction, donc de créer un lien. Cela en supprime une certaine quantité. De même, si l'objet considéré comme source dans un échange est une source de lumière, il n'y aura transmission d'énergie qu'à la première itération.

Il faut de plus que ce critère tienne compte de la cohérence de la hiérarchie des liens. En effet, en retour de récursion de raffinement, pour un noeud donné, si quelques fils ont déjà été établis, il est nécessaire d'établir les autres. Un lien actif ne peut avoir de fils.

Il serait de plus intéressant de pouvoir généraliser ce critère de création des liens. Pour cela, il faudrait arriver à anticiper le comportement du lien aux cours des itérations suivantes. Ce nouveau critère doit être dépendant du critère de raffinement.

4.3.4 Calcul récursif du facteur de forme

Une conséquence de ce raffinement récursif est de nous permettre d'avoir une meilleure approximation des facteurs de forme des liens passifs, ceci en le calculant à partir de ceux de ces fils. En effet, en considérant la formule du facteur de forme entre surfaces, les formules suivantes (e pour l'émetteur et r pour le récepteur) peuvent être démontrées :

$$1. \text{ L'émetteur a été subdivisé : } F_{re} = \sum_{j \in \text{fils}(e)} F_{rj}$$

2. Le récepteur a été subdivisé : $F_{re} = \frac{1}{A_r} \sum_{j \in fils(r)} A_j F_{je}$

Ces formules peuvent aussi être calculées en considérant les énergies reçues. Ainsi, un récepteur reçoit (en considérant l'échange à son niveau) la moyenne des énergies reçues par ces fils, ou la somme des énergies envoyées par les fils de l'émetteur. Dans le cadre de cet échange, il est supposé que la radiosité est uniforme sur les éléments qui y participent. Dans le cadre de milieu non-participatif, nous généralisons les formules précédentes aux calculs des facteurs de forme sur des clusters, et de même, le facteur de forme d'un cluster vers lui-même peut être défini :

$$F_{rr} = \frac{1}{A_r} \sum_{i \in fils(r)} \sum_{j \in fils(r)} A_i F_{ij}$$

Par cette formulation, le lien d'un cluster vers lui-même permet la représentation exacte des échanges internes dans certains cas, comme pour les clusters représentant un ensemble d'éléments dont la visibilité les uns par rapport aux autres est nulle (ensembles de polygones dos à dos, surfaces polygonales convexes, ...) ou un cluster homogène représentant deux surfaces identiques face à face.

Cette formulation permet donc un calcul récursif des facteurs de forme sur les liens passifs, et ainsi une amélioration de la précision de leur calcul.

4.4 Nouvel algorithme

Tous ce qui a été décrit au paragraphe précédent nous permet alors de définir un nouvel algorithme réduisant le nombre de liens stocké. Avec notre nouvelle approche, une itération ne se fait alors plus qu'en deux passes :

1. *RaffinerEtRecolter()* : Etablissement de la hiérarchie des éléments et des liens et calcul de l'irradiance de chaque élément. Cette procédure est décrite plus en détails ci-dessous. Notons tout de même qu'elle suppose que l'irradiance de chaque élément est nulle.
2. *Mise à jour de la représentation hiérarchique (PushPull())* : Cette procédure reste identique à celle décrite plus haut (équation 4.5)

La procédure *RaffinerEtRecolter* est alors un parcours récursif de la hiérarchie des liens. Pendant ce parcours, nous construisons des sous-arbres (raffinement d'un lien actif) ou détruisons des sous-arbres (un lien actif devient passif). Dans la descente (figure 4.2) de la hiérarchie, chaque lien est testé par rapport au critère de raffinement. S'il est au «juste niveau», l'irradiance qu'il transporte est ajoutée au récepteur, puis il est testé par rapport au critère de création pour savoir s'il doit être stocké (si oui, il devient actif). S'il nécessite un raffinement, la descente continue sur ses fils.

En retour de la récursion (figure 4.2) - le lien courant a donc été subdivisé ou bien il a des fils -, la hiérarchie est reconstruite. Ainsi, si aucun des fils n'a été créé, le lien courant est testé par rapport au critère de création pour savoir s'il doit être établi en tant que lien actif. Si des liens-fils existent déjà, il est établi en tant que lien passif. Ce retour sert aussi au calcul récursif de son facteur de forme à partir de ceux des ses liens-fils.

Ce traitement général est à séparer en deux cas, l'un traitant les liens passifs, l'autre traitant les liens actifs et ceux en cours de raffinement (subdivision d'un lien actif). En effet, le parcours de la hiérarchie n'est pas le même pour chacun des liens, puisque pour un lien passif on connaît déjà ses fils, tandis que pour un lien actif, ses fils sont construits par son raffinement, donc par un parcours de la hiérarchie des éléments. En fait, le parcours pour un lien actif est un raffinement classique, tandis que celui pour les liens passifs est un véritable parcours d'une structure d'arbre.

Dans toute cette construction récursive de la hiérarchie, il faut prendre garde à conserver sa cohérence. Ainsi, en retour de récursion, il ne doit y avoir aucun lien créé avant un lien actif. De même, lorsqu'un lien actif devient passif dans la descente, tout le sous-arbre dont il était le noeud doit être détruit.

```

RaffinerEtRecolter(Lien L)
  // Descente
  Si L est au «juste niveau» : // terminaison de la récursion
    Ajout de l'irradiance qu'il transporte au récepteur
  Si L vérifie le critère de création :
    L est stocké en tant que lien actif
  Si non :
    Pour tous les fils l de L faire RaffinerEtRecolter(l)
  // Retour
  calcul du facteur de forme de L à partir de ceux de ses fils
  Si il n'y a pas de lien fils de L :
    Si L vérifie le critère de création :
      L est stocké en tant que lien actif
    Si non :
      L est établie en tant que lien passif
      les liens-fils non-crées sont établies en tant que liens actifs

```

FIG. 4.2: Raffiner et Récolter

4.5 Résultats

Cet algorithme a été testé sur les scènes d'étude¹ du chapitre précédent, ceci afin de vérifier le gain réel en mémoire.

En comparant le nombre de liens que notre algorithme stocke au nombre de liens utiles (figure 4.3), nous pouvons constater que notre critère nous permet de ne conserver que le nombre de liens utiles pour l'itération courante. Les variations constatées sont dues au fait qu'en gardant tous les liens pour le calcul du nombre utile, les facteurs de forme n'étaient pas les mêmes que lorsqu'il y a destruction : ils ont une meilleure approximation. Le nombre de liens utiles représente alors le meilleur des cas. Cette variation due à la recréation se constate sur la scène 4.6, car c'est celle qui comporte le plus de créations de liens (figure 4.9). Mais de manière globale, le nombre de liens actifs reste proche du nombre de liens utiles. De plus, le nombre de liens total suit de près la courbe du nombre de liens actifs. Le surcoût due à la hiérarchie des liens (stockage supplémentaire de liens passifs) est donc bien limité.

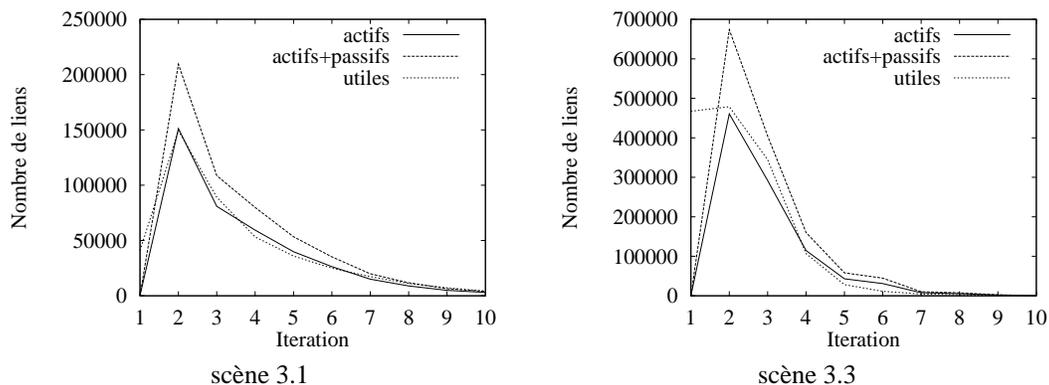


FIG. 4.3: Bilan des liens stockés et comparaison avec le nombre de liens utiles

¹ Il n'y aura pas des résultats sur la scène 3.2, à cause de problèmes de stabilité rencontrés lors des tests

En étudiant le gain en nombre de lien (*figure 4.4*), nous pouvons économiser jusqu'à 50% des liens (sur nos tests). En fait notre critère élimine tous les liens provenant des sources de lumières. plus celle-ci ont une importance des sources dans la scène (énergie qu'elles fournissent par rapport à l'énergie moyenne). Plus une scène sera lumineuse, plus notre gain sera conséquent 4.6.

Algorithme	Classique	Nouveau	Gain
scène 3.1	212848	209433	1,1%
scène 3.3	1100641	673705	38,8 %
scène 4.6	387624	185682	52,1 %

FIG. 4.4: Nombre de liens stocké

Au point de vue gain de mémoire, avec le critère définie ci-dessus, une comparaison à été faite entre la scène 4.6 (car ayant beaucoup de sources, bien moins de liens vont être créé par rapport à l'algorithme classique) et la scène 3.1 (une seule source donc moins de gain théorique en place mémoire). Comme cela avaient été déjà annoncé au chapitre précédent, notre critère permettra un gain d'autant plus grand que les sources lumineuses étaient importantes. Mais même dans le cas de la scène du troupeau de vaches 3.1 (beaucoup de polygones pour une seule source lumineuse), le gain n'est pas négligeable. Il peut même dans des cas extrême atteindre 40%.

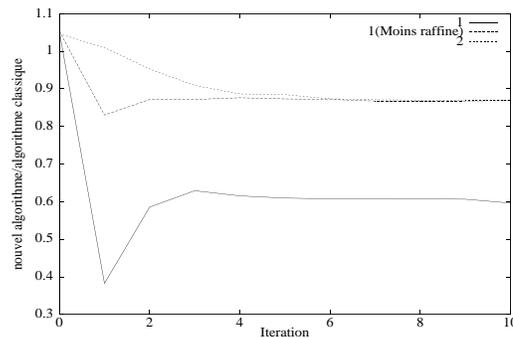


FIG. 4.5: Comparaison de la mémoire utilisée au cours des itérations sur deux cas extrêmes :
1 = scène 4.6 : très forte importance des sources
2 = scène 3.1 : faible importance des sources



FIG. 4.6: Hall très éclairé
84 polygones
1 cluster
16 sources

Algorithme	Classique	Nouveau	Gain
scène 3.1	64,4 Mo	56,0 Mo	13%
scène 3.3	95,0 Mo	72,8 Mo	23%
scène 4.6	25,5 Mo	15,7 Mo	38%

FIG. 4.7: Mémoire utilisée sur nos scènes d'études

Algorithme	Classique	Nouveau
scène 3.1	39mn36s	27mn48s
scène 3.3	2h58mn50s	2h14mn26s
scène 4.6	13mn08s	16mn41s

FIG. 4.8: Temps de calcul pour 4 itérations

Il est intéressant de noter que le temps de calcul (*figure 4.8*) reste proche de celui, d'un algorithme classique. Il est d'autant plus avantageux, que la quantité de liens que l'on recrée à chaque itération est faible. (*figure 4.9*).

4.6 Avantages et inconvénients

Notre nouvel algorithme, en plus du gain en mémoire, offre d'autres avantages que nous allons détailler ici. Nous allons également nous attarder sur ses désavantages.

Itération	scène 3.1	scène 3.3	scène 4.6
4	212	3247	591
5	0	0	0
6	0	0	1445
7	0	0	677

FIG. 4.9: Nombre de liens recréés

De plus, en même temps que ce stage, une idée similaire par l'idée de base (utilisation de la radiosité à transmettre) à été développé au Computer Graphics Group de l'université de Erlangen-Nuremberg[MSSS98]. Nous ferons une comparaison des choix effectués.

4.6.1 Avantages

L'algorithme que nous avons introduit possède un certains nombre d'avantages que nous pouvons résumer en quatre points.

1. Nous n'utilisons que le nombre de liens nécessaires à chaque itération.
2. Nous basculons le nombre de liens du côté des plus forts échanges.
3. Nous améliorons la qualité de l'approximation des facteurs de forme.
4. Nous avons un gain jusqu'à 40% de mémoire tout en gardant un temps de calcul comparable à l'algorithme classique.

Les quatre premiers points proviennent de notre choix de conserver la hiérarchie des liens. En effet, cette structure apporte plusieurs avantages :

- le stockage de tous les liens du raffinement nous permet de ne *considérer que les liens nécessaires à l'échange courant*.
- *Une meilleure approximation des facteurs de forme* : celui-ci peut être calculé récursivement à partir de ceux calculés pour les liens fils. Ceci est particulièrement important pour les clusters, car un approximation du facteur de forme du cluster vers lui-même est toujours différente de zéro, alors qu'avec notre calcul, il peut devenir nul. différents de zéro La valeur ainsi calculée étant plus souvent plus justes, cela a aussi tendance à diminuer le nombre de liens nécessaires.
- le calcul du facteur de forme n'est pas à refaire pour les liens passifs.

De plus, si le temps de calcul reste dans les limites de l'algorithme classique (*figure 4.8*), c'est que non seulement cette structure permet un accès plus rapide aux liens à raffiner (au lieu de parcours de la hiérarchie des éléments à la recherche des liens), mais il n'y a aussi que deux parcours par itérations, l'un sur les liens (RaffinerEtRecolter), l'autre sur la hiérarchie des éléments. Cela compense le coût de la recréation de liens.

Enfin le gain sera d'autant plus grand que les sources auront une forte influence dans la scènes, car notre simple critère ne fait que empêcher le stockage des liens provenant des sources.

4.6.2 Inconvénients

Un désavantage de cet algorithme, c'est que la destruction de liens va parfois demander de leur recalcul. Mais comme le montrent nos études sur nos scènes de référence (*figure 4.9*), ce nombre reste assez faible par rapport au nombre de liens total (*figure 4.3*). C'est la scène 4.6 (Hall simple) qui demande le plus de recalcul, notamment à cause du fait qu'il y a un décalage d'une itération entre l'éclairage du sol et des murs (illumination directe) et celui du plafond (illumination indirecte) et que l'énergie qui circule est moins vite absorbée que dans une scène complexe comme le hall complexe (3.3). Un faible nombre de liens stocké peut demander alors beaucoup de recalcul dans les itérations suivantes. Il y a là un dilemme mémoire-temps de calcul.

4.6.3 Comparaisons

Comme cela a été dit en introduction, le Computer Graphics Group de l'université de Erlangen - Nuremberg a développé un algorithme basé lui aussi sur la radiosité à transmettre. Cependant, les choix suivants ont différé.

Leur solution est de créer un cache d'un nombre fixé de liens. Elle a pour principal avantage de permettre une gestion de la mémoire à utiliser, ce que nous n'avons pas. Cependant, ce cache pourrait être intégré dans notre critère de création, et combiner ainsi les avantages des deux algorithmes.

De plus, la solution du cache peut créer trop de liens car elle ne considère la création de liens que sur le nombre maximal à stocker. Elle n'a pas ce phénomène de bascule des liens vers les lieux de plus fort transfert, car il n'y a pas de destruction de liens inutiles. De plus, s'il y a une bonne gestion de la mémoire, le temps lui sera de plus en plus dépendant du cache considéré (à cause des créations). Notre algorithme offre un compromis entre temps et mémoire.

4.7 Conclusion

Notre nouvel algorithme comporte quatre choix principaux : l'utilisation de la radiosité à transmettre, le stockage de la hiérarchie des liens pour un raffinement récursif, un critère de création de liens et enfin, un calcul récursif des facteurs de forme.

Le cache de liens, solution proposée par le Computer Graphics Group de l'université de Erlangen Nuremberg[MSSS98] pourrait y être ajouté, combinant ainsi les avantages des deux solutions (gestion de la mémoire utilisée par les liens et stockage des seuls liens nécessaires).

De plus, un critère plus évolué nous permettrait d'anticiper l'influence de certains liens pour les itérations futures, ceci pour éviter le stockage inutile de liens. Pour l'instant les seuls liens jamais stockés sont ceux qui proviennent des sources. Mais ceci se ferait au dépend du temps. Avec critère de stockage, les gains en mémoire peuvent être conséquents, et dans tous les cas testés, notre algorithme consomme moins de place mémoire. Il apporte aussi l'avantage de rester proche d'un algorithme classique par le temps de calcul.

Mais pour l'instant, notre algorithme fait que la hiérarchie des éléments est devenue prépondérante au point de vue coût mémoire. Nous allons alors exposer certaines pistes de recherche pour le limiter dans le chapitre suivant.

Chapitre 5

Réduction de la hiérarchie et émission retardée

Le nouvel algorithme obtenu a perdu une part de la complexité de stockage due aux liens. Les résultats obtenus nous montrent que l'influence de la structure de liens restante est minoritaire dans la nouvelle consommation de mémoire. C'est la hiérarchie elle-même qui en utilise la plupart.

Dans un premier temps, nous nous consacrerons à l'élimination des stockages inutiles, cela par une meilleure organisation plus fine de la hiérarchie, mais aussi par une étude déterminant quels sont les paramètres d'une simulation d'éclairage. Ces études ont pour buts de définir la complexité d'une scène à illuminer.

Dans un second temps, nous orienterons vers la réduction de la hiérarchie dans le cadre d'une illumination globale dépendante du point de vue, ce qui permettra certaines approximations

Nous profiterons aussi de ce chapitre pour expliquer un nouvel algorithme en cours de développement, qui en réduisant les valeurs moyennes de radiosité à transmettre, permettrait encore une réduction du nombre de liens.

5.1 Meilleure utilisation de la mémoire

La hiérarchie est une structure qui généralise tout les types d'éléments qui la compose. Cette généralisation entraine un stockage de paramètres inutiles, donc coûteux en place mémoire. De plus, avec notre nouvelle algorithme, nous avons la possibilité de la réduire dans les sous-arbres sans échanges radiatifs. Une meilleure permettrait d'éliminer certains stockages inutiles.

Toujours dans l'optique de la recherche du stockage inutile, nous définirons ce que devrait être un meilleure utilisation de la mémoire.

5.1.1 Représentation de la radiosité

Le cadre de cette étude suppose que le milieu n'est pas participatif. Un cluster n'est alors qu'un noeud de la hiérarchie

Les feuilles de la hiérarchie représentent la base de projection de la fonction de radiosité. Pour notre algorithme, où la notion de radiosité complète n'est plus utile pour les échanges, seules les feuilles doivent alors la stocker. Il en est de même pour l'exitance et la réflectance.

**les valeurs
de radiosité,
de réflectance,
et d'exitance.
ne sont utiles que sur les feuilles de la hiérarchie**

De même, comme la valeur de radiosité des sources reste constante (et égale à leur exitance), tous les éléments due à la subdivision d'une source ne nécessitent pas le stockage de la réflectance (elle est nulle), de la radiosité (c'est l'exitance), et de la radiosité à transmettre. De même, un élément sans exitance (un réflecteur pur) n'aurait pas à stocker une valeur d'exitance. Dans tous les cas, une représentation moins généralisée des éléments permettrait un gain en coût mémoire, car chacun ne stockerait que les paramètres qui lui sont utiles. Le tableau 5.1 récapitule ces divers résultats.

	G	I	ρ	B	ΔB	E
Noeud	•	•		◦	•	
Noeud d'une source	•					•
Feuille généralisée	•	•	•	•	•	•
Feuille d'une Source	•					•
Feuille d'un Réflecteur	•	•	•	•	•	

FIG. 5.1: Nouvelle organisation de la hiérarchie
 $G = \text{Géométrie}$, $I = \text{Irradiance}$, $\rho = \text{réflectance}$
 $B = \text{Radiosité}$, $\Delta B = \text{Radiosité à transmettre}$, $E = \text{Exitance}$

Les marques ◦ ne sont à prendre en compte que pour la radiosité hiérarchique classique, Par contre, il faut éliminer la colonne traitant de la radiosité à transmettre ΔB .

Une première approche de cette idée a été implémentée, en considérant les clusters comme des noeuds de la hiérarchie. Les autres modifications nous demandaient une réorganisation complète de la hiérarchie. Cette simple modification permet d'éviter le stockage de deux vecteurs flottants de dimension 3 (la représentation des spectres lumineux en RGB), soit un gain de 24 octets par clusters. Ainsi sur notre scène très clustérisée (scène 3.1 avec 24338 clusters), l'économie, dès le chargement de la scène, est de 0,55 Mo. Une généralisation à toute la hiérarchie permettrait d'augmenter ce gain au cours des itérations.

5.1.2 Etude de la complexité

Bien qu'en général on ne mesure la complexité d'une scène qu'à travers le nombre de polygones celui-ci n'en est pas le seul paramètre. Il y a aussi l'ensemble des propriétés physiques des polygones (réflectance et exitance), qui peut être plus petit que l'ensemble des polygones dans le cas où plusieurs surfaces partagent les mêmes propriétés physiques, ou plus grand, lorsqu'une texture représente la fonction de réflectance sur une surface. De même, nous pouvons définir des ensembles de propriétés géométriques de la scène, comme l'ensemble des points ou l'ensemble des arêtes. Avec une telle représentation de la scène, il n'y a pas de répétitions de paramètres. Le gain est alors d'autant plus important qu'il y a de répétition dans la scène.

Nous avons implémenté un stockage unique des réflectances et des exitances, par l'introduction de ces paramètres dans un arbre de recherche. Sachant alors qu'un noeud de cet arbre utilise 20 octets (2 pointeurs et 3 flottants pour la représentation RGB), et pour un paramètres répété N fois dans un scène, le gain suit la loi $(2N - 4)/3N$, qui tend rapidement vers 66% de gain de stockage. Ce stockage est avantageux à partir de 3 répétitions d'une même valeur.

5.1.3 Premiers résultats

Les deux structures que nous avons commencé à développer, ont données lieu à une implémentation. Les résultats sont les suivants :

Algorithme	Réduction des liens		Nouvelles modifications	
scène 3.1	56,0 Mo	13,0 % de gain	52,6 Mo	18,3 % de gain
scène 3.3	72,8 Mo	23,0 % de gain	70,0 Mo	26,3 % de gain

Les gains sont intéressants et le serait tant plus avec une organisation encore meilleure que celle qui pour le moment existe.

De plus, avec ces nouvelles structures, nous avons pu simuler l'éclairage d'une scène de 288 000 polygones (figure 5.2), en utilisant 328,5 Mo de mémoire (en 8h34mn sur une Onyx 2 *Infinite Reality*).



FIG. 5.2: Grand Hall

5.2 Vers une hiérarchie réduite

Pour traiter des scènes d'une très grande complexité, nous serons obligés de réduire la hiérarchie décrite ci-dessus. En effet, comme nous étudions l'ont montrés celle-ci consomme une grande partie de la place occupée par un algorithme de radiosit . De plus, une hi rarchie r duite impliquerait une r duction du nombre de liens.

Encore une fois, notre g n ralisation va partir de l' tude du comportement des sources lumineuses. Une fois le transfert de leur  nergie fait dans la premi re it ration, elles n'auront plus aucune influence dans la suite des calculs. Or, toute leur subdivision est conserv e, alors que la seule information «utile» pour le reste de l'algorithme, c'est leur repr sentation. Tout leur raffinement peut  tre alors d truit.

L'utilisation de textures de radiosit s repr sentant le r sultat va nous permettre de g n raliser cela. En consid rant une surface, raffin e en une hi rarchie de profondeur h dont le nombre d' l ments varie entre $1 + 4h$ (dans le meilleur des cas) et $(4^{h+1} - 1)/3$ (pour le pire des cas et pour une structure de Quad-Tree), celle-ci pourrait  liminer cette hi rarchie et ne stocker la repr sentation du r sultat que sous la forme de textures de radiosit  de tailles maximales 4^h . Sachant que le rapport entre un  l ment et une repr sentation en RGB de la radiosit  est d'environ $(1/10)^e$, le gain maximum de stockage tend vers 85% de la m moire actuelle utilis e pour chaque surface (en tenant compte du fait qu'il nous faut stocker deux textures, une pour la radiosit , l'autre pour la radiosit    transmettre). Pour ce r sultat, la formule employ e est :

$$\frac{3}{5} \frac{4^h}{4^{h+1} - 1}$$

Pour le gain minimum, il reste avantageux jusqu'  une profondeur de 3 . La formule dans ce cas est :

$$\frac{1}{5} \frac{4^h}{4h + 1}$$

Ce dernier cas est un cas extr me, c'est un Quad-Tree d g n r . Nous l'avons jamais rencontr  dans nos sc nes tests.

Une autre direction possible serait de faire des calculs de radiosit  d pendants du point de vue, dans l'esprit de l'importance[SAS92]. Cela permettrait de remplacer les clusters par des imposteurs (images avec des informations 3D), afin de limiter la complexit  du calcul. En effet, un cluster   grande distance du point de vue peut, par exemple,  tre remplac  par une surface de polygones textur s. La complexit  deviendrait

alors locale et non plus globale, mais ont perdu un des avantages de la radiosité, qui est l'indépendance de la solution par rapport au point de vue.

Avec une telle approche, nous pourrions envisager un nouvel algorithme qui permettrait le traitement de scènes très complexes en contraignant la mémoire. Il faudrait mettre en place un nouveau raffinement tenant compte du point de vue.

5.3 Un nouvel algorithme : L'émission retardée

Le raffinement est très dépendant de la valeur de radiosité moyenne à transmettre. Dans le but de réduire cette moyenne, toute radiosité inférieure à un critère déterminé est stockée, en attendant qu'elle soit suffisante pour être transmise. Ceci est fait à l'aide d'un filtre \mathbf{H} à définir.

Pour cela une variable supplémentaire (\mathbf{S}_B pour Storage Radiosity) est nécessaire. Cette mise en mémoire ne doit se faire que sur les feuilles de la hiérarchie pour qu'elle soit valable. Cela correspond en fait à faire une compression par ondelettes de Haar. L'algorithme devient alors :

1. Initialisation : (\mathbf{R} désigne l'énergie reçue à une itération. C'est un paramètre purement formel, qui n'est là que pour clarifier l'algorithme)

$$\begin{cases} \mathbf{R}^0 = \mathbf{E}\mathbf{S}_B^0 = 0 \\ \Delta\mathbf{B}^0 = \mathbf{E} \\ \mathbf{B}^0 = \mathbf{E} \end{cases} \quad (5.1)$$

2. Récurrence :

$$\begin{cases} \mathbf{R}^{i+1} = \mathbf{S}_B^i + \rho\mathbf{F}\Delta\mathbf{B}^i \\ \mathbf{S}_B^{i+1} = \mathbf{H}(\mathbf{R}^{i+1}) \\ \Delta\mathbf{B}^{i+1} = \mathbf{R}^{i+1} - \mathbf{S}_B^{i+1} \\ \mathbf{B}^{i+1} = \mathbf{B}^i + \Delta\mathbf{B}^{i+1} \end{cases} \quad (5.2)$$

Ainsi, à la n^{eme} itération :

$$\mathbf{B}^n + \mathbf{S}_B^n = \mathbf{E} + \rho\mathbf{F}(\mathbf{B}^{n-1}) \quad (5.3)$$

Ceci montre bien le rôle joué par la radiosité stockée. La démonstration est simple :

$$\begin{aligned} \mathbf{B}^n &= \sum_{i=0}^n \Delta\mathbf{B}^i \\ &= \mathbf{E} + \sum_{i=1}^n \mathbf{S}_B^{i-1} + \rho\mathbf{F}(\Delta\mathbf{B}^{i-1}) - \sum_{i=0}^n \mathbf{S}_B^i \\ &= \mathbf{E} + \rho\mathbf{F}(\mathbf{B}^{n-1}) - \mathbf{S}_B^n \end{aligned}$$

Attachons-nous maintenant à l'erreur commise avec une telle méthode. Les valeurs de radiosité sans l'erreur commise seront surmontées d'un $\tilde{}$.

$$\begin{aligned} \mathbf{B}^n - \tilde{\mathbf{B}}^n &= \rho\mathbf{F}(\mathbf{B}^{n-1} - \tilde{\mathbf{B}}^{n-1}) - \mathbf{S}_B^n \\ \|\mathbf{B}^n - \tilde{\mathbf{B}}^n\| &\leq \|\mathbf{S}_B^n\| + \|\rho\mathbf{F}\| \|\mathbf{B}^{n-1} - \tilde{\mathbf{B}}^{n-1}\| \\ \varepsilon_n &\leq \alpha_{max} + \|\rho\mathbf{F}\| \varepsilon_{n-1} \\ \varepsilon_n &\leq \|\rho\mathbf{F}\|^n \varepsilon_0 + \sum_{i=0}^{n-1} \|\rho\mathbf{F}\|^i \alpha_{max} \end{aligned}$$

(avec $\forall n, \|\mathbf{S}_B^n\| \leq \alpha_{max}$, α_{max} borne maximum de \mathcal{S}_B :) Or $\varepsilon_0 = 0$, donc

$$\varepsilon_n \leq \frac{1}{1 - \|\rho\mathbf{F}\|} \alpha_{max} \quad (5.4)$$

Cette idée a commencé à être implémentée, mais demande pour être réellement efficace d'utiliser la nouvelle hiérarchie décrite précédemment. En effet, seules les feuilles de la hiérarchie ont besoin de l'information \mathcal{S}_B . Il n'est pas nécessaire non plus, de l'utiliser pour les sources.

Quant à l'erreur commise, des calculs ont été faits en stockant jusqu'à 50% de la radiosité à transmettre moyenne courante, sans erreur visible à la convergence (voir le tableau 5.3). Par contre, avec un tel algorithme, il y a de fortes erreurs aux premières itérations, comme cela se remarque sur les images 5.3. Mais il n'y a pas encore de véritables résultats sur le gain d'un tel algorithme.

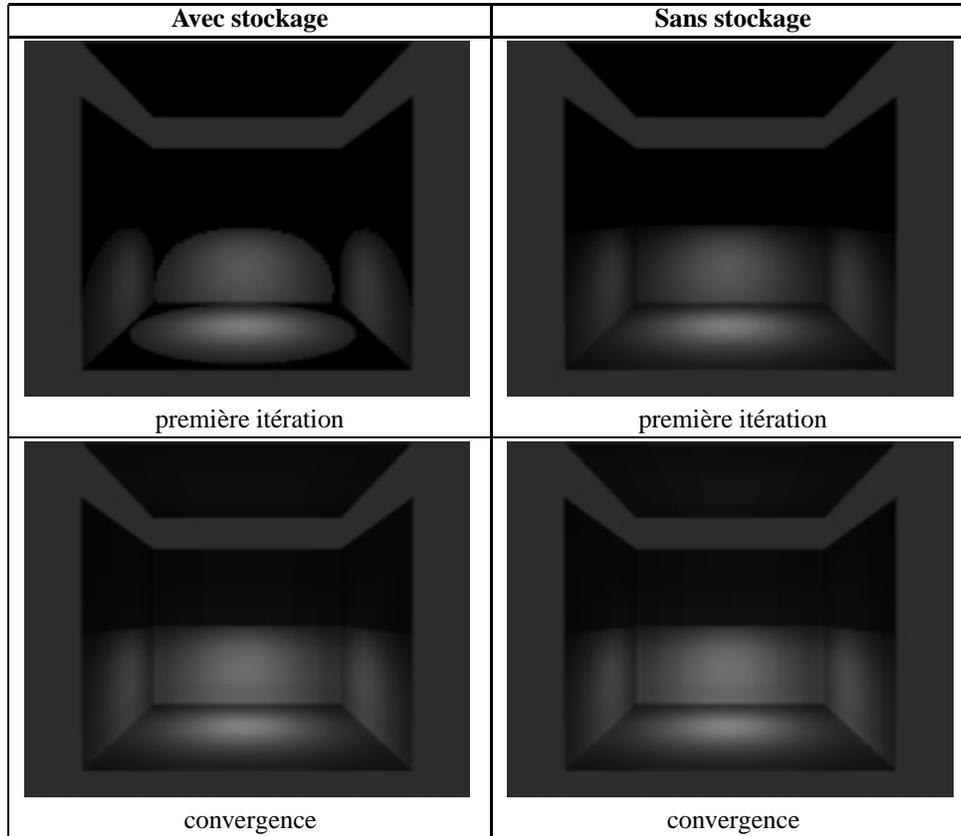


FIG. 5.3: Test de l'émission retardée

5.4 Conclusion

Le nombre de liens stockés une fois réduit, la hiérarchie devient prépondérante dans la consommation de mémoire. C'est pourquoi nous avons présenté une nouvelle approche pour réduire la taille de la hiérarchie elle-même. Elle se décompose en deux points principaux, l'un portant sur une meilleure représentation de la complexité de la scène, l'autre se dirigeant vers la réduction de cette complexité par la création d'une hiérarchie réduite.

De plus, nous avons présenté un nouvel algorithme basé sur une émission retardée, algorithme qui aurait tendance à réduire encore plus le nombre de liens. Il reste à confronter à la réalité, l'intérêt théorique de l'algorithme de radiosit  avec émission retard e. Pour cela, les travaux sur une hi rarchie r duite doivent  tre men    bout.

Chapitre 6

Conclusion

Dans le domaine de l'illumination global, la radiosité hiérarchique apparaît comme une des méthodes les plus rapides pour le calcul d'images réaliste. Elle permet aussi un contrôle d'erreur, ouvrant ses applications à la simulation.

Dans les méthodes de radiosité, basées sur des équations d'échanges radiatifs, la radiosité hiérarchique se distingue par la structuration qu'elle fait des échanges, tenant compte de leur différence d'influence. De plus, le regroupement d'objets (clustering) de même comportement dans un échange, permet de réduire la complexité des scènes face au temps de calcul.

Cependant, cette structuration était gourmande en place mémoire. Pour pouvoir augmenter la taille des scènes simulables, une modification de l'algorithme actuel paraissait nécessaire. Cet objectif nous a conduit à étudier les deux composantes principales de la radiosité hiérarchique, la hiérarchie elle-même (représentation multi-résolution des objets) et les liens (établissant un échanges en deux éléments de la hiérarchie).

6.1 Etude des structures de radiosité hiérarchique

En effet, une telle réduction nous demandait de bien cerner le comportement face à l'utilisation de la mémoire de l'algorithme actuel d'un point de vue théorique et statistique. L'étude menée nous a permis de préciser les lieux de fortes créations de ces structures, situés dans les 3-4 premières itérations, et d'en préciser les causes, une forte valeur de radiosité à transmettre

Une conclusion très intéressante, ignorée jusqu'alors, est que la place mémoire utilisée par la hiérarchie peut être aussi importante, voir même plus grande, que celle occupée par les liens. Cette conclusion est particulièrement vrai pour le cas des algorithmes avec clustering. Cette remarque a motivé les travaux sur la réduction de la consommation de mémoire par la hiérarchie au chapitre 5.

Ces études nous ont aussi permis de définir un nouveau critère d'utilité d'un liens, se basant non-plus sur la radiosité de chaque élément, mais sur la différence entre deux itérations.

6.2 Nouvel algorithme

Les études menées nous ont conduit au développement d'un nouvel algorithme permettant de réduire le nombre de liens stockés. Ses composantes principales sont les suivantes :

La radiosité à transmettre : La radiosité de chaque élément n'est plus recalculé à chaque itération. Elle est obtenue par la sommation des apports de chaque étape. C'est l'idée de base du nouvel algorithme, similaire à celui de la radiosité progressive.

La hiérarchie des liens : Cette nouvelle structure est ici utilisée pour permettre un raffinement récursif, et la destruction des liens inutiles. De plus, le raffinement se fait en parcourant cette structure, et non plus en parcourant la hiérarchie des éléments, ce qui nous apporte un gain en temps de calcul.

Un critère de création des liens : Celui-ci permet d'anticiper les niveaux auxquels les liens seront utiles à l'itération suivante. Combiné à la hiérarchie des liens, cela permet de ne stocker que ces derniers. Pour l'instant, nous ne créons jamais de liens depuis les sources lumineuses.

Un calcul récursif des facteurs de forme : La hiérarchie des liens nous permet de calculer les facteurs de forme des liens à partir de ceux de leurs fils. Nous avons ainsi pu définir un calcul pour le facteur de forme d'un cluster vers lui-même.

Toutes ces modifications nous ont donné un algorithme qui, sur nos tests, a permis jusqu'à un gain en place mémoire de 40%.

De plus, notre algorithme reste pour l'instant un bon compromis entre le gain en place mémoire et le temps de calcul. Il reste dans les limites des temps d'un algorithme classique, et peut être même plus rapide.

Enfin, il permet un calcul plus précis des facteurs de forme, notamment pour les liens d'un cluster vers lui-même.

6.3 Réduction de la mémoire utilisée par la hiérarchie

Une fois cet algorithme développé, le stockage de la hiérarchie est devenue prépondérante dans l'utilisation de la mémoire. Nous avons alors orienté nos travaux vers celle-ci.

Pour commencer, par une meilleure définition de la complexité d'une simulation d'éclairage, et par une réorganisation de la hiérarchie, nous avons pu éliminer une partie du stockage inutile de paramètres. Cela nous a permis de gagner encore jusqu'à 5% par rapport à notre algorithme précédent.

Dans un second temps, nous avons proposée des méthodes permettant la réduction de la hiérarchie par l'emploi de textures de radiosité ou d'imposteurs dans le cadre d'une simulation d'éclairage dépendante du point de vue.

6.4 Perspectives

La suite de nos travaux sera orienté dans deux axes principaux

Emission retardée : Un algorithme retardant l'émission de radiosité trop faible a été développé. Il permettrait, par une atténuation de la radiosité moyenne, de réduire le nombre de liens. Des tests supplémentaires devront être effectués.

Amélioration du critère de création des liens : Nous allons développer ce critère pour ne stocker qu'un nombre liens encore plus petit par une meilleure anticipation des échanges futures. Mais ceci ce fera au dépend du temps. De plus, un cache de lien pourrait aussi y être intégré afin d'avoir une meilleure gestion de la mémoire que cette structure occupe, dans l'esprit du Computer Graphics Group de l'université de Erlangen-Nuremberg[MSSS98].

Hiérarchie réduite : Nous mettrons au point une nouvel algorithme qui permettrait une réduction du stockage nécessaire pour la hiérarchie. Nous nous placeront dans le cadre d'une simulation d'éclairage pour un point de vue fixé, ce qui nous permettra d'utiliser des textures au lieu d'un hiérarchie des subdivisions d'une surface, et des imposteurs à la place certains clusters.

Figures couleur



FIG. 6.1: Un trop grand nombre de liens

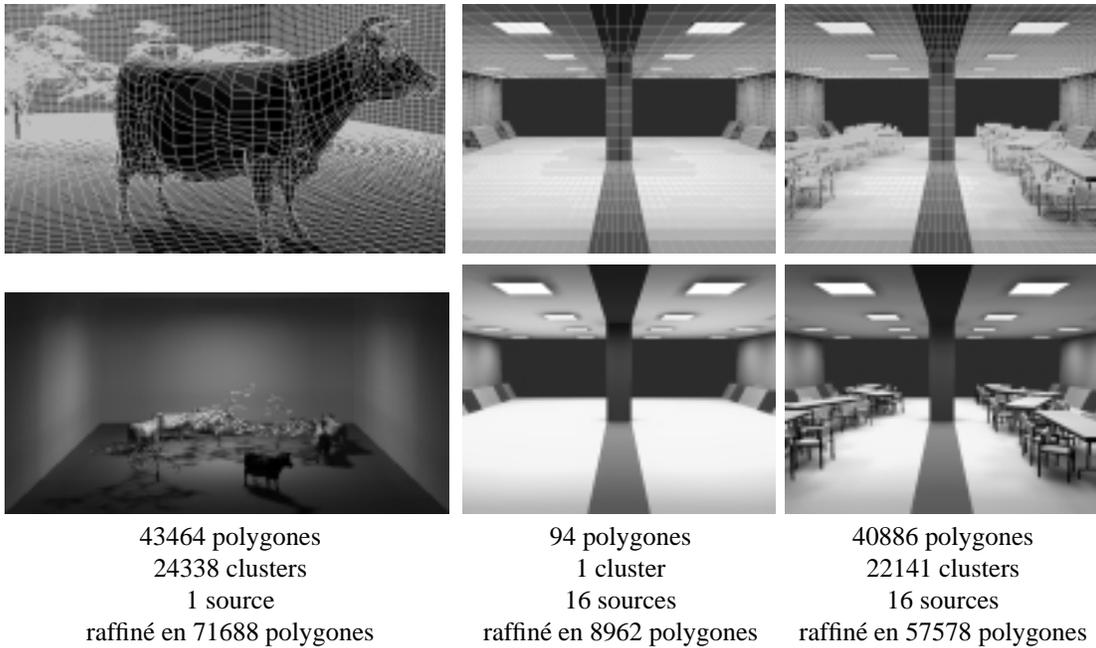


FIG. 6.2: scènes d'étude



FIG. 6.3: Grand Hall

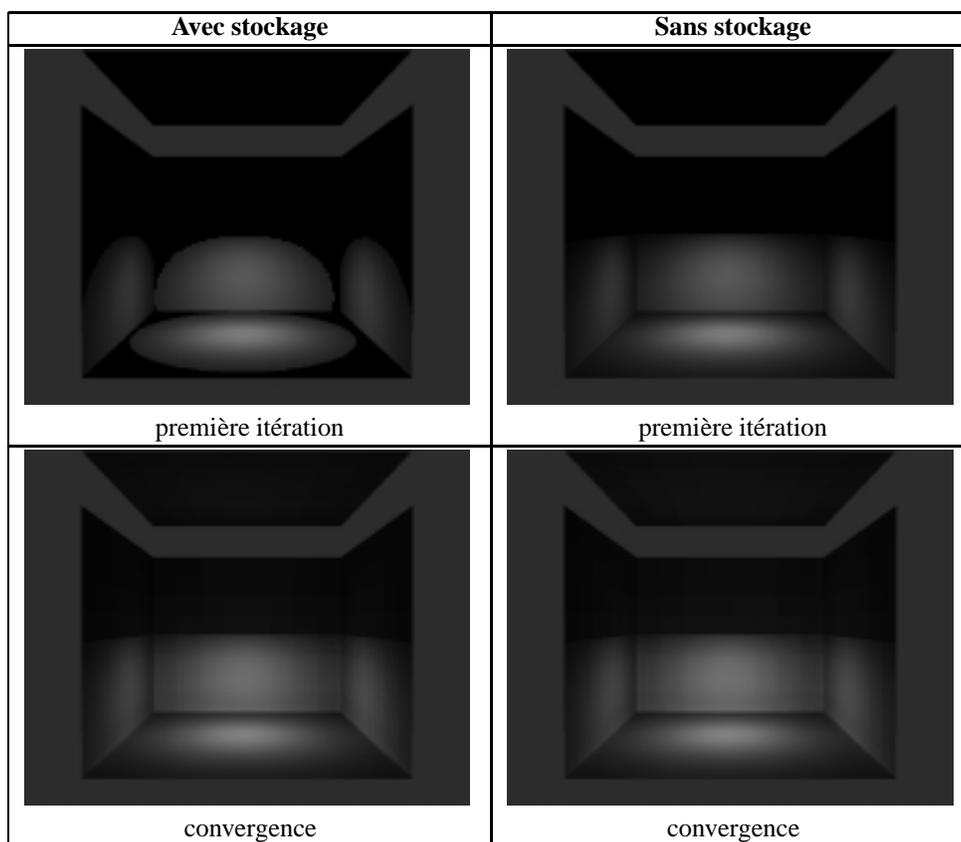


FIG. 6.4: Test de l'émission retardée

Bibliographie

- [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, pages 75–84, August 1988.
- [CGIB86] Michael F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3) :26–35, March 1986.
- [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, January 1997.
- [DS97] George Drettakis and François Sillion. Interactive update of global illumination using A line-space hierarchy. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings, Annual Conference Series*, pages 57–64. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [GH96] S. Gibson and R. J. Hubbard. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5) :297–310, 1996.
- [GH97] Simon Gibson and R. J. Hubbard. Perceptually driven radiosity. *Computer Graphics Forum*, 16(2) :119–128, June 1997.
- [GSH94] Reid Gershbein, Peter Schroder, and Pat Hanrahan. Textures and Radiosity : Controlling Emission and Reflection with Texture Maps. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 51–58, 1994.
- [GTGB84] Cindy M. Goral, Kenneth K. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. volume 18, pages 213–222, July 1984.
- [HOL96] Nicolas HOLZSCHUCH. *Le contrôle de l'erreur dans les méthodes de radiosit  hi rarchique*. Th se de doctorant, Universit  Joseph Fourier, Grenoble -France, 1996.
- [HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. volume 25, pages 197–206, July 1991.
- [LSG94] D. Lischinski, B. Smits, and D. P. Greenberg. Bounds and error estimates for radiosity. *Computer Graphics*, 28(Annual Conference Series) :67–74, July 1994.
- [MSSS98] M. Stamminger, H. Schirmacher, Ph. Slusallek, and H.-P. Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Forum (Eurographics '98 Proceedings)*, 17(3), 1998.
- [SAG94] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [SAS92] Brian E. Smits, James R. Arvo, and David H. Salesin. An Importance-Driven Radiosity Algorithm. In *Computer Graphics (ACM SIGGRAPH '92 Proceedings)*, volume 26, pages 273–282, July 1992.

- [SDS95] Francois Sillion, George Drettakis, and Cyril Soler. A Clustering Algorithm for Radiance Calculation in General Environments. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 196–205, New York, NY, 1995. Springer-Verlag.
- [Sha97] Erin Shaw. Hierarchical radiosity for dynamic environments. *Computer Graphics Forum*, 16(2) :107–118, 1997.
- [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.
- [SP94] François X. Sillion and Claude Puech. *Radiosity & Global Illumination*. Morgan Kaufmann, San Francisco, California, 1994.
- [WH97a] Andrew Willmott and Paul Heckbert. An empirical comparison of progressive and wavelet radiosity. In Julie Dorsey and Phillip Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 175–186, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.
- [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, April 1997. <http://www.cs.cmu.edu/radiosity/emprad-tr.html>.