

UNIVERSITÉ JOSEPH FOURIER-GRENOBLE 1
SCIENCES ET GÉOGRAPHIE
U.F.R. D'INFORMATIQUE ET DE MATHÉMATIQUES APPLIQUÉES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER

Discipline : Informatique

arrêtés ministériels du 5 juillet 1984 et du 30 mars 1992

Présentée et soutenue publiquement
par

Cyril SOLER

Le 18 Décembre 1998

**REPRÉSENTATIONS HIÉRARCHIQUES DE LA VISIBILITÉ
POUR LE CONTRÔLE DE L'ERREUR
EN SIMULATION DE L'ÉCLAIRAGE**

effectuée sous la direction de

François SILLION

au sein du laboratoire GRAVIR-IMAG. UMR CNRS C5527.

COMPOSITION DU JURY :

Pierre-Jean	LAURENT	Président
Bernard	PEROCHE	
Claude	PUECH	
Dani	LISCHINSKI	Rapporteur
Hans-Peter	SEIDEL	Rapporteur
François	SILLION	Directeur

A mes parents

Remerciements

Je voudrais tout d'abord remercier Claude Puech pour m'avoir accueilli dans son équipe et pour l'attention qu'il porte chaque jour au bien être de chacun.

Je remercie tout particulièrement Pierre-Jean Laurent d'avoir accepté de présider mon jury de soutenance, ainsi que Bernard Péroche pour sa participation. Merci enfin à Dani Lischinski et Hans-Peter Seidel, pour le soin qu'ils ont porté à l'examen de mon travail, et notamment à la lecture d'une version partiellement anglaise de mon document.

Je n'aurai eu qu'un seul directeur de thèse dans toute ma vie mais c'était le bon. J'ai eu énormément de plaisir à travailler avec François Sillion, autant sur le plan purement informatique qu'humain.

L'équipe iMAGIS possède une renommée internationale pour la qualité de ses résultats, mais également pour son excellente ambiance de travail collectif. Sans tous les membres d'iMAGIS qui m'ont aidé, ce travail ne serait pas ce qu'il est. Je voudrais remercier tout particulièrement Fredo, Niko et Eric pour leur disponibilité, leur bonne humeur et leur gentillesse, ainsi que pour leur recette de l'eau sucrée à la verveine (Non Bichon, je ne préfère pas de la Chartreuse !).

Mes parents sont toujours à mes cotés pour m'aider en toutes circonstances. Je ne les remercierai jamais assez.

Je dédie mes lettrines à celle qui me manque chaque jour, et qui fut ma première passion à rayures, ainsi que la partie droite de la figure 8.18 à *zoé*, qui n'a toujours soutenu du bout de sa langue fourchue. Je dédie ma collection de *core dump*, patiemment accumulée, au créateur du C++, rappelant ainsi la mémoire du pointeur nul, si souvent malmené. Merci également à L^AT_EX pour m'avoir appris le contrôle absolu de mon système nerveux, et au Professeur Halimi pour le contrôle de mon système endocrinale.

Enfin, puisque l'on garde en général le meilleur pour la fin, je dois avouer que je dois à peu près tout à ma Clairette, qui est une perle parmi les perles.

Table des matières

Introduction	15
I Contrôle de l'erreur dans la méthode de radiosité hiérarchique	17
1 Étude de la radiosité hiérarchique	19
1.1 Principe de la méthode de radiosité	19
1.1.1 Définitions	19
1.1.2 Équilibre des échanges lumineux	22
1.1.3 Discrétisation	24
1.1.4 Résolution de l'équation discrète	25
1.2 Méthodes hiérarchiques de Radiosité	29
1.2.1 Principe	30
1.2.2 Généralisation de la hiérarchie : le <i>clustering</i>	33
1.2.3 Expression matricielle	34
1.2.4 Convergence	35
1.2.5 Rendu de la solution	36
2 Contrôle de l'erreur en radiosité hiérarchique	39
2.1 Caractérisation de l'erreur	39
2.1.1 Modélisation	40
2.1.2 Discrétisation	41
2.1.3 Calcul des coefficients du système linéaire	41
2.1.4 Résolution du système	43
2.1.5 Visualisation de la solution	43
2.1.6 Conclusion	43
2.2 Impact de l'approximation des facteurs de forme	44
2.3 Approche <i>a posteriori</i>	44
2.3.1 Obtention des bornes sur le facteur de forme	45
2.3.2 Convergence des itérations	46
2.3.3 Utilisation des bornes <i>a posteriori</i>	46
2.4 Approche <i>a priori</i>	47
2.4.1 Outils mathématiques	47
2.4.2 Conditionnement de la matrice de radiosité	48
2.4.3 Influence de l'erreur sur la solution	49
2.4.4 Extension à la radiosité hiérarchique	50
2.4.5 Utilisation pratique	51
2.5 Discussion et conclusion	51

II	Calcul contrôlé des facteurs de forme	55
3	Contrôle de l'erreur d'approximation de la visibilité dans le calcul des facteurs de forme	57
3.1	Définitions	57
3.1.1	Expression du facteur de forme en terme de mesures de droites	58
3.2	Contrôle de l'erreur et information multi-échelles de visibilité	59
3.3	Approximation du facteur de forme	60
3.4	Combinaison des facteurs de forme	61
3.4.1	Formule de combinaison	61
3.4.2	Application au calcul récursif contrôlé	62
3.5	Algorithme de calcul récursif du facteur de forme	63
3.6	Calcul et stockage de l'information de visibilité	66
3.6.1	Calcul	66
3.6.2	Stockage	67
3.7	Conclusion	67
4	Approximation contrôlée des facteurs de forme en dimension deux	69
4.1	La radiosité en dimension 2	69
4.1.1	Définitions	70
4.1.2	Équation d'équilibre	72
4.1.3	Calcul des facteurs de forme	72
4.2	Approximations des facteurs de forme	74
4.2.1	Bornes basées sur l'emploi de normes usuelles	74
4.2.2	Bornes basées sur la géométrie des obstacles	75
4.3	Estimation de la corrélation	77
4.4	Résultats	79
4.5	Conclusion	80
5	Approximation contrôlée des facteurs de forme en dimension trois	81
5.1	Encadrement du facteur de forme	81
5.2	Discussion et conclusion	84
III	Approche perceptive de l'erreur de visibilité	87
6	Génération d'ombres à partir de données fréquentielles	89
6.1	Conservation de l'auto-corrélation	89
6.1.1	Résultats	91
6.1.2	Conclusion	92
6.2	Échantillonnage par fréquence	92
6.2.1	Principe	92
6.2.2	Application à la génération des ombres	93
6.3	Conclusions	95
7	Calcul d'ombres par convolution	97
7.1	État de l'art	98
7.1.1	Méthodes d'échantillonnage	98
7.1.2	Méthodes interactives	98
7.1.3	Utilisation de structures auxiliaires	99
7.1.4	Ombres et méthodes d'éclairage global	99
7.1.5	Méthodes multi-échelles	100
7.1.6	Position de notre travail	100
7.2	Principe de la méthode de convolution	100

7.2.1	Cas d'école : objets plans et parallèles	100
7.2.2	Cas général	103
7.3	Extensions directes	105
7.3.1	Sources non uniformes	105
7.3.2	Sources volumiques	106
7.3.3	Récepteur non polygonal	106
7.3.4	Source a l'infini	107
7.3.5	Obstacle semi-transparent	108
7.3.6	Éclairage direct	109
7.4	Mise en œuvre	109
7.4.1	Du choix de la géométrie virtuelle	110
7.4.2	Calcul des images pour la convolution	111
7.4.3	Calcul de la convolution	115
7.4.4	Rendu des textures	117
7.5	Contrôle de l'erreur	120
7.5.1	Discussion qualitative des sources d'erreur	120
7.5.2	Réduction de l'erreur	121
7.5.3	Formule de combinaison	125
7.5.4	Estimation perceptive de l'erreur d'approximation	127
7.5.5	Algorithme de contrôle de l'erreur	129
7.6	Résultats	131
7.7	Conclusion	133

IV Synthèse 137

8	Radiosité hiérarchique avec convolution	139
8.1	Motivation	140
8.2	Principe	141
8.3	Propagation cohérente de l'énergie	142
8.3.1	Calcul des transferts	142
8.3.2	<i>Push/pull</i>	142
8.4	Contrôle du raffinement	143
8.4.1	Principe	143
8.4.2	Stratégie de raffinement	145
8.4.3	Algorithme de raffinement	147
8.4.4	Critère pour le raffinement standard	149
8.4.5	Paramètres du raffinement	149
8.4.6	Influence des paramètres de raffinement	150
8.5	Rendu	153
8.5.1	Algorithme d'affichage	153
8.5.2	Visibilité intra-récepteur	154
8.5.3	Le lissage de Gouraud	155
8.5.4	Cas des environnements texturés	158
8.5.5	Optimisation du rendu	159
8.6	Résultats	160
8.7	Conclusion	162

Conclusion 165

A	Quelques résultats concernant les harmoniques sphériques	169
A.1	Définitions et propriétés	169
A.1.1	Polynômes de Legendre	170
A.1.2	Polynômes de Legendre associés	170
A.1.3	Le théorème d'addition	171
A.1.4	Harmoniques sphériques	171
A.1.5	Harmoniques sphériques pour une fonction réelle	172
A.2	Évaluation efficace d'une combinaison linéaire d'harmoniques sphériques .	173
A.3	Rotation des harmoniques sphériques	174
A.3.1	La formule de rotation	175
A.3.2	Décomposition efficace d'une combinaison linéaire d'harmoniques sphériques après rotation	176
A.3.3	Coût de l'algorithme	177
A.3.4	Applications au <i>clustering</i> directionnel	178
B	Notes sur la conjecture d'unimodalité	181
B.1	Unimodalité de la convolution	181
B.2	Expression de l'irradiance	185
	Bibliographie	187

Table des figures

1.1	Coordonnées polaires pour un vecteur	19
1.2	Définition de l'angle solide	20
1.3	Notations pour la radiance et la réflectance	21
1.4	Notations pour le noyau de la radiosité	23
1.5	Exemple de discrétisation de la géométrie	24
1.6	Notations pour la formule analytique du facteur de forme point-polygone	27
1.7	Méthode de l'hémicube	27
1.8	Surface subdivisée pour la réception uniquement	30
1.9	Raffinement d'un lien	31
1.10	Subdivision des quadrilatères et des triangles	31
1.11	Principe du <i>Push/Pull</i>	32
1.12	Exemples de hiérarchies de clusters	33
1.13	Contribution d'un lien à la matrice de radiosité hiérarchique	35
1.14	Convergence de la radiosité hiérarchique	36
1.15	Principe du <i>smooth shading</i>	37
1.16	Exemples de <i>quadtree</i>	37
2.1	Erreur de corrélation noyau-visibilité	42
3.1	Exemples d'obstacle isotropes et anisotropes	61
3.2	Importance de la corrélation entre les sous-obstacles	63
3.3	Pseudo-code pour le calcul du facteur de forme	64
3.4	Calcul récursif avec répartition des seuils d'erreur	65
4.1	Exemple de scène bidimensionnel	70
4.2	Repérage des droites du plan	71
4.3	Courbe croisée et enveloppe convexe.	71
4.4	Configuration générale de facteur de forme	72
4.5	Définition des courbes croisées et non croisées	73
4.6	Exemple d'information de visibilité	75
4.7	Configuration entre segments	75
4.8	Exemple d'information multi-échelles de visibilité (1)	77
4.9	Exemple d'information multi-échelles de visibilité (2)	78
4.10	Configuration expérimentale	79
4.11	Temps CPU et sous-obstacles visités	80
5.1	Paramétrisation des droites de l'espace	82
5.2	Calcul et encadrement de la zone d'intégration	83
5.3	Analyse statistique d'un obstacle	84
6.1	Synthèse d'une texture d'illumination par FFT	90
6.2	Images de référence (1)	91
6.3	Résultats	91

6.4	Méthode d'échantillonnage fréquentiel	93
6.5	Image de test pour l'échantillonnage fréquentiel	94
6.6	Résultats de l'échantillonnage fréquentiel	94
7.1	Définition des ombres progressives	97
7.2	Configuration idéale pour la convolution	101
7.3	Cas où source, obstacle et récepteur sont plans et parallèles	101
7.4	Exemples de calcul d'ombre par convolution	103
7.5	Construction de la géométrie virtuelle pour le cas général	104
7.6	Utilisation de la géométrie virtuelle pour une scène 3D	105
7.7	Exemple de sources non uniforme	106
7.8	Exemple de source volumique	107
7.9	Récepteur non polygonal	108
7.10	Source à l'infini (notations)	108
7.11	Eclairage direct par convolution	110
7.12	Source et récepteur utiles	111
7.13	Calcul <i>off-screen</i> de l'image du l'obstacle virtuel	112
7.14	Cônes de projections	113
7.15	Aliassage de l'image de la source	114
7.16	Mesure de l'influence de l'anti-aliassage sur la précision	115
7.17	Effet de l'anti-aliassage de l'image de l'obstacle	115
7.18	Temps de calcul des FFT	116
7.19	Vitesses de calcul des convolutions par l'extension <i>OpenGL</i>	116
7.20	Zone de sécurité pour la convolution	117
7.21	Exemples de maillages d'affichage	118
7.22	Calcul des coordonnées de texture	119
7.23	Artefacts de subdivision du récepteur	122
7.24	Subdivision de l'obstacle, du récepteur ou de la source	123
7.25	Subdivision de la source (1)	124
7.26	Subdivision de la source (2)	125
7.27	Effet de la corrélation entre les sous-obstacles	125
7.28	Situations extrêmes pour la corrélation	126
7.29	Notations pour la construction du critère d'erreur	128
7.30	Algorithme de contrôle de l'erreur	130
7.31	Subdivision de l'obstacle	131
7.32	Exemple de calcul d'ombres par convolution (1)	132
7.33	Exemple de calcul d'ombres par convolution (2)	133
7.34	Temps de calcul des scènes d'exemple	133
8.1	Substitution d'une convolution à un maillage complexe	141
8.2	Schéma de principe du <i>Push/Pull</i>	143
8.3	Pseudo-code pour le <i>Push/Pull</i> avec textures	144
8.4	Construction d'un <i>shaft</i> entre deux boîtes englobantes	146
8.5	Algorithme de sélection des obstacles potentiels	147
8.6	Algorithme de raffinement avec liens de convolution	148
8.7	Influence du type de lien sur le temps de calcul	151
8.8	Influence qualitative des paramètres de raffinement (1)	152
8.9	Influence qualitative des paramètres de raffinement (2)	153
8.10	Influence qualitative des paramètres de raffinement (3)	154
8.11	Influence qualitative des paramètres de raffinement (4)	155
8.12	Influence qualitative des paramètres de raffinement (5)	156
8.13	Algorithme de rendu de la solution	156
8.14	Simulation de l'ombre auto-portée	157
8.15	Discontinuités d'affichage (1)	157

8.16	Discontinuités d'affichage (2)	158
8.17	Algorithme de rendu avec textures de réflectance	159
8.18	Rendu avec textures de réflectance	159
8.19	Comparaison avec la radiosité hiérarchique classique	161
8.21	Maillage de la solution figure 8.20	161
8.20	Exemple de scène à visibilité complexe	162
8.22	Mélange de maillage et de textures	162
A.1	Notations relatives aux harmoniques sphériques	169
A.2	Repérage d'une direction par deux angles	171
A.3	Mode de calcul efficace des harmoniques sphériques	173
A.4	Pseudo-code pour le efficace des valeurs des harmoniques sphériques	174
A.5	Décomposition d'un changement de repère en rotations	175
A.6	Calcul des nouveaux coefficients après une transformation quelconque.	178
A.7	Vitesse de rotation des harmoniques sphériques	179
B.1	Notations pour le théorème d'unimodalité	182
B.2	Notations	184

Introduction

La simulation de l'éclairage, et plus généralement les méthodes de calcul des échanges radiatifs, interviennent dans de nombreux domaines comme l'architecture, la réalité virtuelle, les jeux vidéo, la simulation botanique, le séchage infra-rouge, les réseaux de télécommunication, l'acoustique. Appliquée à la réalité virtuelle et aux jeux vidéo, la simulation de l'éclairage permet d'obtenir des images réalistes. En architecture, urbanisme, acoustique ou en télécommunication par ondes radio, la simulation d'environnements complexes offre la possibilité d'étudier les caractéristiques optimales du modèle sans payer le prix d'une étude expérimentale sur le terrain. L'intérêt de la simulation devient évident dans des domaines où une telle étude est impossible. C'est le cas de la simulation botanique, qui prend en compte la quantité de lumière reçue par les plantes pour en modéliser la croissance sur plusieurs années.

Les méthodes de simulation de l'éclairages ont cependant un coût de calcul qui n'est pas négligeable. Dans toutes ces applications, ce coût apparaît comme un facteur limitant, et ne peut être réduit que par le biais d'approximations diverses, au prix d'une diminution de la qualité du résultat. La recherche de ce compromis entre la qualité et la rapidité d'une simulation est ce que nous appelons le contrôle de la précision.

Dans les différents domaines d'application que nous avons cité, la position de ce compromis n'est pas la même : en réalité virtuelle, ou dans les jeux vidéo, nous nous contentons en effet d'un calcul grossier pour pouvoir remplir des conditions d'interactivité. A l'opposé, pour une simulation précise de l'éclairage dans une pièce, en vue de choisir le revêtement des murs ou le type des lampes en fonction d'effets subtils de l'éclairage, nous fournirons un effort de calcul important.

La notion même de *qualité de la simulation* intervenant dans le contrôle de la précision diffère d'une application à l'autre. Au cinéma, par exemple, une grande importance est attachée à l'aspect visuel de la simulation en fonction de critères humains : une scène éclairée avec des lampes supplémentaires mais invisibles ne choque pas le spectateur, alors qu'une balle projetant une ombre carrée ne passera pas inaperçue. Dans d'autres domaines, en revanche, les artefacts visuels n'ont aucune importance et la notion de qualité admet une définition différente : pour une simulation botanique physiologique, connaître précisément la quantité totale de lumière reçue par chaque feuille d'une plante importe plus que d'éviter les discontinuités d'une représentation discrète de l'éclairage.

Une des qualités essentielles demandées à une méthode de contrôle de la précision est la possibilité de proposer à son utilisateur un positionnement continu, entre une simulation précise et longue à calculer, et une solution approximative obtenue en un temps interactif. En architecture, par exemple, on peut avoir besoin d'une estimation rapide de l'éclairage dans une pièce pour régler la position des meubles, puis effectuer une simulation plus précise pour observer l'impact sur l'ambiance lumineuse, des matériaux et luminaires utilisés.

Dans une simulation de l'éclairage par la *méthode de radiosité*, le coût total du calcul est constitué à plus de 90% de l'évaluation des relations de visibilité entre les objets. Le contrôle de la précision au détriment du temps de calcul passe donc inévitablement par des approximations de la visibilité. Nous nous posons alors la question de savoir prévoir

l'impact de telles approximations sur la qualité d'une simulation de l'éclairage. Négliger les phénomènes de visibilité crée tout d'abord des erreurs locales, comme par exemple la disparition d'une ombre. Les méthodes de simulation de l'éclairage modélisent cependant l'équilibre global des interactions lumineuses. De ce fait, une approximation de la visibilité entre deux objets répercute une certaine erreur, surplus ou manque de lumière, dans toute la scène. Dans ce travail, nous avons tenté d'apporter des réponses à la question du contrôle de la précision en simulation de l'éclairage, par approximation de la visibilité : nous avons tout d'abord étudié l'impact de ce type d'approximation, en vue de connaître par avance la latitude qu'ont les algorithmes de calcul de transferts d'énergie locaux dans l'usage de ces approximations. Nous avons ensuite défini de tels algorithmes, capables de transformer au mieux, dans les limites fixées, du temps de calcul en précision.

Dans une première partie, nous poserons les bases d'algorithmes performants de simulation de l'éclairage, ou *algorithmes hiérarchiques de calcul de la radiosité*. Nous étudierons ensuite plus particulièrement, l'impact des approximations de la visibilité sur la précision des images. Nous verrons tout d'abord comment une mesure purement analytique de l'erreur peut conduire à un contrôle mathématique de la précision. Nous établirons pour cela un lien entre la qualité, mesurée mathématiquement, d'une solution de radiosité et les approximations des transferts d'énergie intervenant dans les méthodes hiérarchiques de radiosité. Associée à un algorithme de contrôle de l'erreur sur le calcul de chaque transfert d'énergie, cette approche rend alors possible le calcul d'une solution répondant à des critères de précision quantitatifs. Nous présenterons dans la deuxième partie de ce travail de tels algorithmes, basés sur une formulation hiérarchique de la visibilité.

Nous savons cependant que les erreurs auxquelles la perception humaine est sensible ne peuvent pas être caractérisées par une simple distance mathématique entre une solution calculée et une solution de référence, notamment parce qu'une telle approche ne détecte pas les mêmes types d'erreur que l'œil humain. Le contrôle de la précision selon des critères perceptifs de qualité doit en effet faire appel à une mesure spécifique de l'erreur.

Nous proposons dans une troisième partie, plusieurs approches du contrôle des approximations de la visibilité, basées sur une mesure perceptive de l'erreur. Nous nous intéresserons plus particulièrement à la synthèse artificielle des ombres, à partir de caractéristiques statistiques ou morphologiques des obstacles qui les créent. Nous présenterons notamment une méthode de synthèse des ombres, basée sur l'utilisation de l'opération de convolution et qui a l'avantage de produire des images implicitement dépourvues d'artefacts visuels.

Les méthodes hiérarchiques de radiosité utilisent généralement une seule approche du contrôle de la qualité pour simuler à la fois la répartition globale de l'énergie lumineuse et des détails d'ombres très fins. Comme nous venons de le voir, ces deux aspects d'une simulation de l'éclairage requièrent des algorithmes de contrôle de la précision de différente nature. Nous présenterons dans une dernière partie, comme synthèse de nos travaux, un algorithme hiérarchique de radiosité prenant en compte cette nécessité. Dans ce nouvel algorithme, la répartition globale de l'énergie lumineuse est simulée par une méthode classique de contrôle de la précision, alors les ombres fines, qui influent peu sur l'équilibre final de l'énergie lumineuse et qui nécessitent en revanche un traitement adapté, sont calculées grâce aux méthodes présentées à la partie III. Nous verrons que l'algorithme obtenu est capable de calculer très rapidement des simulations de l'éclairage de haute qualité visuelle, et offre un réglage efficace du temps de calcul en fonction de la précision.

Première partie

**Contrôle de l'erreur dans la
méthode de radiosité
hiérarchique**

Chapitre 1

Étude de la radiosit  hi rarchique

AVANT de pr senter dans ce chapitre le principe des m thodes hi rarchiques de radiosit , nous commencerons par d finir diff rents termes et quantit s habituellement utilis s en simulation de l' clairage. Nous  tablirons ensuite une relation traduisant l' quilibre des  changes d' nergie lumineuse dans une sc ne. L' quation obtenue sera r solv e gr ce   la m thode de radiosit  que nous  tendrons ensuite   sa version hi rarchique.

1.1 Principe de la m thode de radiosit 

La m thode de radiosit  est une m thode d' l ments finis de type *Galerkin* appliqu e   l' quation d' quilibre des  changes lumineux. En voici le principe.

1.1.1 D finitions

Cette section pr sente quelques notions et notations essentielles attach es   la repr sentation de la lumi re et   son interaction avec l'environnement.

(a) Rep rage des rayons lumineux

Dans l'air et dans le vide, la lumi re se propage en ligne droite. Les trajectoires des rayons lumineux sont donc repr sent es par des droites de \mathbb{R}^3 , et leur direction de propagation par des vecteurs de \mathbb{R}^3 , que nous rep rerons par leurs coordonn es polaires, comme indiqu  sur la figure 1.1.

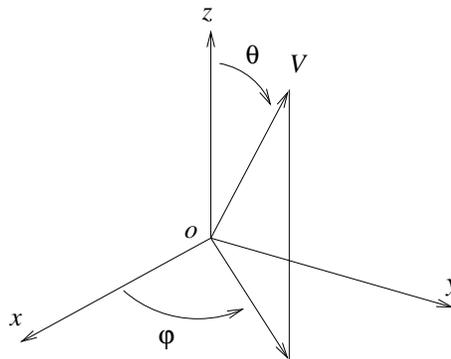


FIG. 1.1: Coordonn es polaires pour un vecteur v de \mathbb{R}^3 . Les angles θ et φ appartiennent respectivement   $[0, \pi]$ et $[0, 2\pi]$

(b) Angle solide

La notion d'*angle solide* est une généralisation à la dimension trois de la notion d'angle :

Définition 1 L'angle solide ω sous lequel on voit une surface S d'un point P est l'aire de la projection de S sur la sphère unité centrée en P .

Cette définition admet une analogie en dimension deux, où l'angle θ sous lequel on voit une surface L à partir d'un point P peut également être défini comme la longueur de la projection de L sur le cercle unité de centre P . Cette analogie est représentée sur la figure 1.2.a¹. Un angle solide infinitésimal $d\omega$ représente donc une fraction infinitésimale de rayons lumineux issus d'un point.

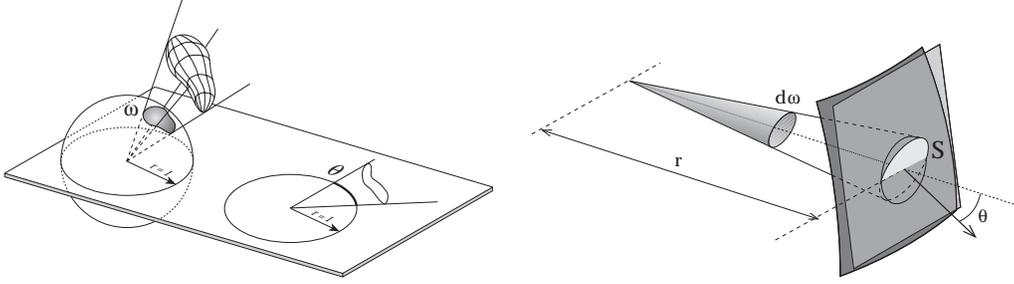


FIG. 1.2: (a) L'angle solide ω défini comme la mesure de la projection d'un objet sur une sphère unité en deux et en trois dimensions. (b) Approximation de l'angle solide pour une petite surface. Voir note de bas de page.

Lorsque δS est une petite surface d'aire δA , située à une distance r du point P , suffisamment grande, l'angle solide $d\omega$ sous lequel on voit δS depuis P peut être asymptotiquement approché par (C.f figure 1.2.b) :

$$d\omega = \frac{\delta A \cos \theta}{r^2} \quad (1.1)$$

(c) Radiance

L'énergie lumineuse est désignée sous le terme de *radiance* dont la définition s'énonce ainsi :

Définition 2 On appelle radiance la quantité d'énergie lumineuse $L(x, \theta, \varphi)$ se propageant en un point x de l'espace dans une direction (θ, φ) donnée, par unité de temps, d'angle solide, et d'aire perpendiculaire à la direction de propagation.

L'énergie lumineuse associée à un pinceau lumineux d'angle solide $d\omega$ traversant une surface d'aire dx pendant un temps dt , et dans la direction (θ, φ) vaut donc

$$L(x, \theta, \varphi) \cos \theta dx d\omega dt$$

où $\cos \theta dx$ est l'aire de dx projetée dans la direction caractérisée par (θ, φ) . (C.f figure 1.3.a)

(d) Réflectance

Dans le cas le plus général, un pinceau de lumière de radiance $L_i(x, \theta, \varphi)$, d'angle solide $d\omega$ arrivant en un point x d'une surface S se réfléchit dans toutes les directions du demi-espace tangent à S en x suivant la loi de réflectivité (C.f figure 1.3.b) :

$$L(x, \theta_0, \varphi_0) = \rho_g(x, \theta_0, \varphi_0, \theta, \varphi) L_i(x, \theta, \varphi) \cos \theta d\omega \quad (1.2)$$

¹Les figures 1.2, 1.3 et 1.4 sont extraites du livre de François Sillion et Claude Puech, *Radiosity and Global Illumination* [SP94]

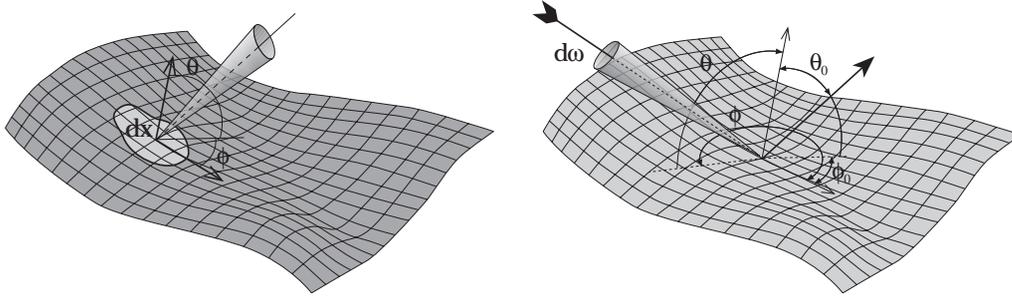


FIG. 1.3: (a) Incidence d'un rayon lumineux sur une surface. (b) notations pour la réflectance. Voir note de bas de page, page 20.

Suivant le type du matériau rencontré, la fonction de réflectance ρ peut être assez complexe. On distingue cependant deux cas très simples : les surfaces idéalement spéculaires et les surfaces idéalement diffuses.

Définition 3 Une surface S est dite idéalement spéculaire si elle ne renvoie de la lumière que dans la direction de réflexion donnée par la loi de Descartes ($\theta_0 = \theta, \phi_0 = \phi \pm \pi$). On a alors :

$$L(x, \theta_0, \phi_0) = \rho_g(x) L_i(\theta_0, \phi_0 + \pi [2\pi]) \cos \theta_0$$

Une telle surface se comporte donc comme un miroir parfait. Ce modèle de réflectance à été essentiellement utilisé dans les méthodes de lancer de rayon [Whi80, GAC⁺88, HH84].

Définition 4 Une surface est dite idéalement diffuse de réflectivité $\rho(x)$, en x , si un pinceau de lumière arrivant en x est réfléchi uniformément dans toutes les directions du demi-espace tangent à la surface en ce point.

Dans ces conditions, la loi de réciprocité, qui stipule que la lumière se comporte indépendamment de son sens de propagation impose alors que l'intensité lumineuse réfléchie est également indépendante de la direction d'incidence :

$$L(x, \theta_0, \phi_0) = \rho_g(x) L_i(x, \theta, \phi) \cos \theta d\omega$$

Il est alors d'usage de poser :

$$\rho(x) = \pi \rho_g(x) \quad (1.3)$$

La quantité $\rho(x)$ est appelée réflectance diffuse de S en x . On a entre autres

$$\rho(x) \in [0, 1]$$

De nombreux domaines de la simulation de l'éclairage utilisent ce modèle de réflectance. Nous l'utiliserons également tout du long de ce document, sauf cas particulier expressément signalé. Dans le cadre de la simulation de l'éclairage par la méthode de radiosit , que nous introduirons plus loin, cette hypoth se est souvent accompagn e d'une autre hypoth se selon laquelle les sources de lumi re  mettent uniform ment dans toutes les directions. Dans la r alit , les surfaces purement diffuses sont plut t rares (Peintures mates, papier,...).

Dans le cas o  toutes les surfaces sont id alement diffuses et toutes les sources  mettent uniform ment dans toutes les directions, il est facile de montrer que la radiance en un point x d'une quelconque surface de la sc ne, ne d pend plus de la direction (θ_0, ϕ_0) [SP94]. Nous la noterons donc $L(x)$.

(e) Radiosité

Les termes de *radiosité*, *irradiance* et *émittance* définis dans cette section désignent dans différents contextes, l'énergie lumineuse totale arrivant ou repartant d'un point x :

Définition 5 On appelle *radiosité* la puissance lumineuse totale émise par unité de surface en un point x , intégrée sur toutes les directions du demi-espace Ω tangent à la surface considérée en x :

$$b(x) = \int_{\Omega} L(x, \theta_0, \varphi_0) \cos \theta_0 d\omega$$

La radiosité s'exprime donc en Watt.m^{-2} . Dans le cas de surfaces idéalement diffuses, un calcul simple montre que l'on peut écrire [SP94] :

$$b(x) = \pi L(x) \quad (1.4)$$

La radiosité rend par ailleurs compte de la quantité de lumière perçue par l'oeil. C'est donc la quantité qui est généralement calculée en simulation de l'éclairage.

Définition 6 On appelle *irradiance* la puissance lumineuse totale par unité de surface arrivant en un point x , intégrée sur toutes les directions du demi-espace Ω tangent à la surface considérée en x :

$$b(x) = \int_{\Omega} L_i(x, \theta_0, \varphi_0) \cos \theta_0 d\omega$$

Lorsque la radiosité est directement émise par la surface, plutôt que simplement réfléchie, on parle alors d'*émittance*. Par analogie avec l'équation (1.4), cette quantité, notée $e(x)$ est calculée à partir de la radiance émise $L_e(x)$ au point x par :

$$e(x) = \pi L_e(x) \quad (1.5)$$

1.1.2 Équilibre des échanges lumineux

Nous allons maintenant établir une équation traduisant l'équilibre des échanges d'énergie lumineuse dans un environnement dont on connaît la réflectance idéalement diffuse et l'émittance en tout point.

(a) Équation d'équilibre

Plaçons nous en un point x d'une surface S . La radiance issue de x peut être décomposée en deux termes : un terme d'émission propre $L_e(x)$, qui est non nul si x est situé sur une source de lumière ; un terme correspondant à la radiance $L_i(x, \theta, \varphi)$ issue de pinceaux lumineux provenant de toutes les directions incidentes en x à la surface et qui se réfléchissent en x , selon la loi (1.2). Cette égalité s'exprime donc comme :

$$L(x) = L_e(x) + \rho_g(x) \int_{\Omega} L_i(x, \theta, \varphi) \cos \theta d\omega \quad (1.6)$$

La radiance $L_i(x, \theta, \varphi)$ provenant nécessairement d'un autre point y de la scène (Cf figure 1.4 pour les notations), et étant uniforme le long d'un rayon lumineux, nous pouvons écrire que :

$$L_i(x, \theta, \varphi) = L(y)$$

L'intégrale intervenant dans l'équation (1.6) peut alors être reformulée en sommant sur tous les points y de la scène :

$$L(x) = L_e(x) + \rho_g(x) \int_{y \in S} L(y) v(x, y) \cos \theta d\omega \quad (1.7)$$

Les points x et y ne se faisant pas automatiquement face, on introduit la fonction $v(x, y)$, valant 1 ou 0 selon si x et y sont, ou ne sont pas, mutuellement visibles.

Grâce aux relations (1.1) et (1.3), nous obtenons :

$$L(x) = L_e(x) + \rho(x) \int_{y \in \mathcal{S}} L(y) \frac{\cos \theta \cos \theta'}{\pi r^2} v(x, y) dy$$

Puis, grâce à (1.4) et (1.5), nous trouvons :

$$b(x) = e(x) + \rho(x) \int_{y \in \mathcal{S}} \underbrace{b(y) \frac{\cos \theta \cos \theta'}{\pi r^2}}_{k(x, y)} v(x, y) dy \quad (1.8)$$

Cette équation, appelée *équation de radiosité* traduit l'équilibre des échanges lumineux dans une scène : la radiosité $b(x)$ issue de x est égale à l'émission $e(x)$ en x plus la radiosité issue de la contribution de tous les autres points de la scène visibles depuis x , pondérée par un terme géométrique $k(x, y)$ prenant en compte la distance entre x et y ainsi que l'orientation relative des surfaces qui les portent. Le terme $k(x, y)$ est également appelé *noyau de la radiosité* en (x, y) .

Connaissant l'emplacement des sources de lumière (*i.e* la fonction $x \mapsto e(x)$) et la géométrie de la scène, cette équation définit implicitement la radiosité $b(x)$ en tout point x de la scène.

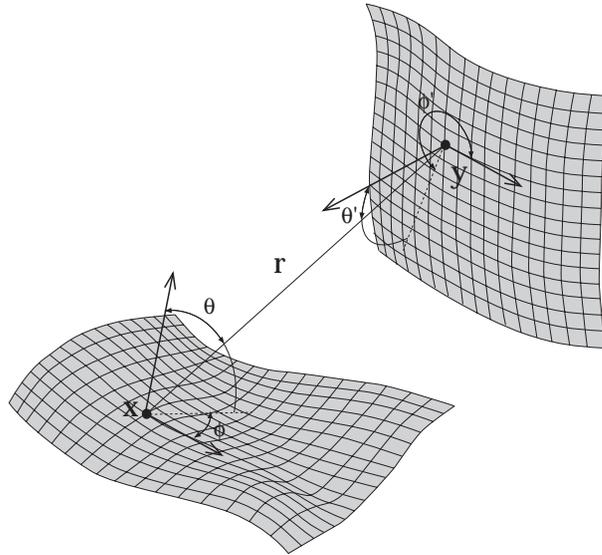


FIG. 1.4: Distance et angles associés à un couple de points mutuellement visibles. Voir note de bas de page, page 20

(b) Traitement de la couleur

Nous n'avons pas jusqu'à maintenant parlé de couleurs. L'équation (1.8) n'est en effet valable que pour une longueur d'onde donnée et toutes ses quantités sont définies par unité de longueur d'onde. Pour obtenir une simulation en couleur, il suffit de discrétiser le spectre de lumière visible et résoudre l'équation de radiosité pour chaque longueur d'onde choisie. La nouvelle équation se décompose donc en autant d'équations que de longueurs d'ondes considérées. Elle peut être également regardée comme une seule équation dont l'inconnue est une fonction à valeurs dans \mathbb{R}^n , et dans laquelle le produit entre le vecteur de réflectances et le vecteur des intensités correspondant à chaque longueur d'onde s'effectue composante à composante. Cette modélisation suppose cependant qu'il n'y a

aucun transfert d'énergie entre les différentes longueurs d'onde (phénomène appelé *fluorescence*), ce qui permet de conserver l'indépendance des équations pour les différentes longueurs d'onde. Une telle supposition ne limite cependant pas notablement la modélisation d'effets de transferts de couleurs (*color bleeding*) [GTGB84], puisque la couleur des surfaces, *i.e* leur réflectance, est la plupart du temps composée de plusieurs des longueurs d'onde de base.

Sur les écrans d'ordinateurs, les couleurs sont rendues par mélange de trois longueurs d'onde de base : le rouge, le vert et le bleu. On résout donc l'équation de radiosité, dans la plupart des implémentations, pour une fonction à valeur dans \mathbb{R}^3 . La solution est donc en chaque point un triplet de valeurs (*rouge, vert, bleu*).

1.1.3 Discrétisation

Excepté dans de très rares cas d'école [SP94], on ne connaît pas de solution analytique à l'équation *intégrale* (1.8). On approche généralement la solution de cette équation par une méthode de collocation [AC97] ou une méthode d'éléments finis, dont fait partie la *méthode de radiosité*.

La méthode de radiosité est plus précisément une *méthode de Galerkin*. Elle consiste donc à approcher la solution b par un élément \bar{b} d'un sous-espace vectoriel des fonctions continues par morceau sur S , de dimension finie. On choisit généralement \bar{b} polynomiale par morceaux. Nous présentons ici brièvement la démarche de discrétisation correspondant à des polynômes de degré 0. Bien que des travaux aient été réalisés en utilisant des polynômes de degré supérieur [Zat93, TM93], cette approximation est la plus couramment utilisée.

Notons E l'ensemble des fonctions uniformes par morceaux sur S . Plus précisément, nous pavons S par des sous-surfaces A_i d'aire non nulle et partageant leur frontière avec leurs voisines, et approchons la solution b par une fonction uniforme sur chaque A_i . La figure 1.5 donne un exemple d'une telle discrétisation.

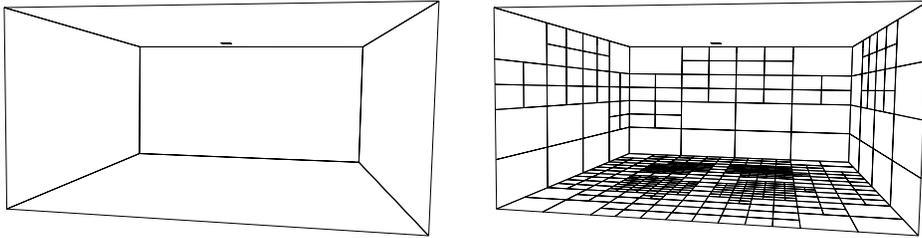


FIG. 1.5: Exemple de scène discrétisée. Une scène constituée de 6 polygones (à gauche) a été discrétisée en 525 surfaces élémentaires sur lesquelles la radiosité est approchée par une valeur uniforme.

Notons a_i l'aire de A_i et posons

$$\varphi_i(x) = \frac{1}{\sqrt{a_i}} \mathbf{1}_{A_i}(x) = \begin{cases} \frac{1}{\sqrt{a_i}} & \text{si } x \in A_i \\ 0 & \text{sinon} \end{cases}$$

Alors $(\varphi_k)_{k \in N}$ est une famille libre orthonormée pour le produit scalaire

$$\langle f, g \rangle = \int_S f(x)g(x)dx$$

Soit $\bar{b}(x) = \sum_{i=1}^N B_i \varphi_i(x)$ la meilleure approximation de $b(x)$ dans E au sens des moindres carrés. Nous avons alors

$$\forall i \in \mathbb{N}_n \quad \langle b - \bar{b}, \varphi_i \rangle = 0$$

i.e

$$\forall i \in \mathbb{N}_n \quad B_i = \int_{A_i} b(x) dx$$

En appliquant cette relation aux deux membres de l'équation (1.8), nous obtenons :

$$B_i = \underbrace{\frac{1}{a_i} \int_{A_i} e(x) dx}_{\text{Noté } E_i} + \frac{1}{a_i} \int_{A_i} \int_S b(y) v(x, y) \frac{\cos \theta \cos \theta'}{\pi r^2} dx dy$$

Les A_i ne forment pas une partition de S , mais à part au niveau des frontières entre ces éléments, qui forment un ensemble de mesure nulle, tout point y est contenu dans exactement une surface A_j . Nous avons donc

$$B_i = E_i + \frac{1}{a_i} \int_{A_i} \sum_{j=1}^N \int_{A_j} b(y) v(x, y) \frac{\cos \theta \cos \theta'}{\pi r^2} dx dy$$

D'autre part, sur A_j , la radiosité s'écrit

$$b(x) = B_j + \Delta B_j$$

où ΔB_j désigne l'écart de la radiosité à sa moyenne sur A_j .

Si b est continue sur A_j , alors l'approximation $b(x) \# B_j$ est asymptotiquement "exacte" quand le diamètre de A_j tend vers 0². Nous pouvons donc asymptotiquement approcher la solution de l'équation de radiosité continue par l'élément de E dont les composantes B_i vérifient la relation suivante :

$$B_i = E_i + \frac{1}{a_i} \int_{A_i} \sum_{j=1}^N \int_{A_j} B_j v(x, y) \frac{\cos \theta \cos \theta'}{\pi r^2} dx dy$$

i.e :

$$\forall i \in \mathbb{N}_N \quad B_i = E_i + \rho_i \sum_{j=1}^N B_j F_{ij} \quad \text{avec} \quad F_{ij} = \frac{1}{a_i} \int_{A_i} \int_{A_j} v(x, y) \frac{\cos \theta \cos \theta'}{\pi r(x, y)^2} dx dy \quad (1.9)$$

Les quantités F_{ij} appelées *facteurs de forme*, traduisent la contribution de chaque surface A_j à la radiosité de la surface A_i . Pour obtenir les valeurs B_i , il suffit donc de former et résoudre le système linéaire (1.9), ce que nous expliquons dans la section suivante.

1.1.4 Résolution de l'équation discrète

Pour former le système linéaire (1.9), il faut tout d'abord calculer les valeurs des facteurs de forme F_{ij} . En (a) nous donnerons plusieurs méthodes de calcul des facteurs de forme auxquelles nous nous référerons tout au long de ce document. En (b), nous traiterons de la résolution du système proprement dit.

²La convergence est alors en norme infinie. Lorsque l'aire a_i tend vers 0, elle n'est qu'en norme L_p

(a) Calcul des facteurs de forme

Lorsque v est uniforme sur $A_i \times A_j$, les surfaces A_i et A_j sont dites *totalelement invisibles* lorsque $v \equiv 0$ et *totalelement visibles* lorsque $v \equiv 1$. Ces deux situations sont les seules dans lesquelles on connaît une expression analytique du facteur de forme entre A_i et A_j . Dans les premier cas, il est nul, dans le second, une formule analytique, complexe mais utilisable en pratique, donne sa valeur [SH93]. Cette expression analytique, que nous ne donnerons pas ici, est obtenue à partir de la formulation suivante du facteur de forme basée sur l'intégration le long des contours C_i et C_j des surfaces A_i et A_j :

$$F_{ij} = \frac{1}{2\pi a_i} \oint_{C_i} \oint_{C_j} \log r(x, y) \vec{dx} \cdot \vec{dy}$$

Cette expression est elle même obtenue en appliquant la formule de Stokes à la forme présentée sur l'équation (1.9) du facteur de forme. Notons que le fait même de déterminer si un couple de surfaces appartient à une des deux catégories de visibilité uniforme nécessite des calculs géométriques exacts [HW91a, TH93].

Les formules analytiques étant néanmoins numériquement coûteuses, on leur préfère souvent l'approximation grossière suivante du facteur de forme, qui est d'autant plus justifiée que les surfaces A_i et A_j sont relativement éloignées :

$$F_{ij} \# a_j \frac{\cos \theta \cos \theta'}{\pi d^2}$$

Dans cette expression, d désigne la distance entre les deux surfaces, mesurée par exemple entre leurs centres respectifs.

Le facteur de forme F_{ij} peut également être considéré comme la valeur moyenne sur A_i du *facteur de forme point-polygone*, noté F_{x, A_j} d'un point x de A_i vers A_j :

$$F_{ij} = \frac{1}{a_i} \int_{A_i} F_{x, A_j} dx \quad \text{avec} \quad F_{x, A_j} = \int_{A_j} \frac{\cos \theta \cos \theta'}{\pi r^2} dy \quad (1.10)$$

Lorsque la visibilité est uniformément égale à 1 sur le domaine d'intégration, une formule analytique simple permet d'évaluer F_{x, A_j} , lorsque A_j est un polygone :

$$F_{x, A_j} = \sum_{k=1}^p n \cdot \gamma_k \quad \text{avec} \quad \gamma_k = \alpha_i \frac{R_i \wedge R_{i+1}}{\|R_i \wedge R_{i+1}\|} \quad (1.11)$$

Les quantités γ_i , R_i sont définies sur la figure 1.6. Le vecteur unitaire n est la normale à la surface contenant x .

Lorsque la visibilité est partielle, seuls une intégration numérique fine ou des calculs géométriques poussés permettent d'obtenir des valeurs exactes du facteur de forme. L'intégration peut être conduite sur $A_i \times A_j$ en échantillonnant $k(x, y)$, ou bien sur A_i uniquement, en échantillonnant F_{x, A_j} . Dans le second cas on calcule analytiquement F_{x_k, A_j} sans tenir compte de la visibilité, puis on le multiplie par une valeur moyenne de la visibilité de x_k vers A_j elle même obtenue par échantillonnage de $v(x, y)$:

$$F_{ij} \# \frac{1}{a_i} \sum_k V_k F_{x_k, A_j} \quad \text{avec} \quad V_k = \frac{1}{N} \sum_{p=0}^N v(x_k, y_p)$$

Dans tous les cas, la visibilité rendant la fonction à intégrer discontinue, l'utilisation des méthodes d'intégration à base d'interpolation polynomiale (Quadrature de Gauss, formules de Newton-Cotes) n'apporte rien à la précision du résultat. On utilise plutôt des méthodes de type *méthode des rectangles* avec une précision d'échantillonnage suffisante.

L'intégrale définissant F_{x, A_j} peut également être approchée de façon efficace grâce à la technique de l'*hémicube* [CG85], consistant à pré-calculer les facteurs de forme d'un point

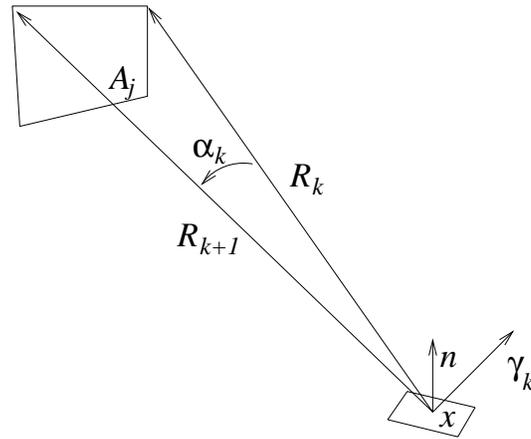


FIG. 1.6: Notations pour la formule du facteur de forme point-polygone.

situé au centre d'un cube vers des éléments de subdivision des faces d'un demi-cube, dont la base est centrée en ce point (C.f figure 1.7). Pour calculer le facteur de forme d'un point x vers une surface A_j , on projette A_j sur l'hémicube et on ajoute la contribution précalculée de toutes les facettes du cube dont le centre est dans la projection de A_j . Cette technique a l'avantage de permettre de traiter les cas de visibilité partielle : il suffit pour cela de projeter également les obstacles potentiels entre x et A_j et de retirer les facettes masquées par cette nouvelle projection.

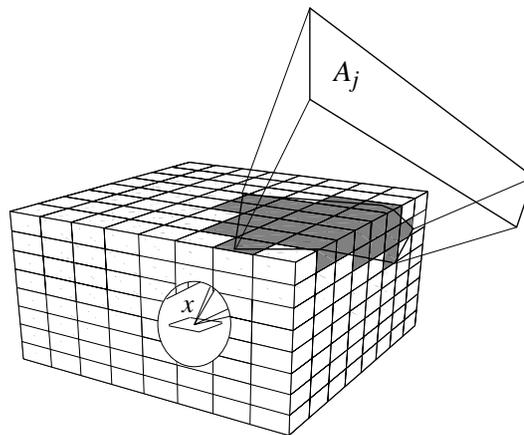


FIG. 1.7: Illustration de la méthode de l'hémicube.

Enfin, certaines méthodes sont basées sur l'énumération des relations géométriques liant la source, le récepteur et les éléments faisant occlusion pour en déduire une expression exacte du facteur de forme point-polygone ou même polygone-polygone [DDP97]. Des méthodes analytiques permettent également de se situer à mi-chemin, en offrant de calculer exactement le terme de visibilité moyenne V_k intervenant dans l'intégration numérique du facteur de forme [Tam92, Hec92, DF94]. Ces méthodes sont cependant difficiles à mettre en œuvre et très sensibles à la précision numérique.

Le calcul des facteurs de forme est donc en général une opération coûteuse et approximative. En formant le système linéaire, il est cependant possible de faire l'économie du calcul d'un certain nombre de F_{ij} , en se basant sur les propriétés suivantes, qui découlent rapidement de l'expression intégrale du facteur de forme :

Propriété 1 Si A_i et A_j sont coplanaires, alors $F_{ij} = 0$.

Propriété 2 On a toujours $a_i F_{ij} = a_j F_{ji}$.

Enfin, nous feront grand usage de la propriété suivante, qui traduit la conservation de l'énergie dans la scène :

Propriété 3 Pour toute facette A_i , la somme des facteurs de forme vers toutes les autres facettes A_j de la scène vérifie :

$$\sum_{j \neq i} F_{ij} \leq 1$$

En effet, nous pouvons écrire en tout point x de A_i , que :

$$\sum_{j \neq i} F_{x,A_j} = \int_S \frac{\cos \theta \cos \theta'}{\pi r^2} v(x,y) dy$$

Notons Ω l'ensemble des directions incidentes en x . En utilisant (1.1) on obtient :

$$\begin{aligned} \int_S \frac{\cos \theta \cos \theta'}{\pi r^2} v(x,y) dy &\leq \int_{\Omega} \frac{\cos \theta}{\pi} d\omega \\ \text{et} \quad \int_{\Omega} \frac{\cos \theta}{\pi} d\omega &= \frac{1}{\pi} \int_{C(x,1)} ds \\ &= 1 \end{aligned}$$

Dans cette dérivation, $C(x,1)$ désigne le cercle de centre x et de rayon 1 dans le plan de A_i . L'élément $\cos \theta d\omega$ peut en effet être lu comme l'aire de la projection d'un élément de surface infinitésimal de la sphère unité centrée en x , sur le plan de A_i . L'intégrale vaut donc l'aire du cercle C , i.e π .

Enfin, $\sum_{j \neq i} F_{ij}$ est la moyenne sur A_i de $\sum_{j \neq i} F_{x,A_j}$. D'où le résultat.

(b) Résolution itérative

Notons M la matrice $N \times N$ de terme général

$$m_{ij} = \rho_i F_{ij}$$

Le système (1.9) s'écrit alors

$$B = E + MB \quad (1.12)$$

où B est le vecteur des valeurs de radiosité B_i et E celui des émittances E_i .

Ce système admet une unique solution à la condition nécessaire et suffisante que la matrice $I - M$ soit inversible, i.e [Atk89] que le rayon spectral de M vérifie

$$\rho(M) < 1$$

Notons S_{∞} la norme matricielle engendrée par la norme $\|\cdot\|_{\infty}$ de \mathbb{R}^N :

$$S_{\infty}(M) = \sup_{x \neq 0} \frac{\|Mx\|_{\infty}}{\|x\|_{\infty}} = \max_i \sum_i |m_{ij}|$$

On a alors pour toute matrice A [Atk89] :

$$\rho(A) \leq S_{\infty}(A)$$

D'autre part

$$\begin{aligned}
S_\infty(M) &= \max_i \sum_j |m_{ij}| \\
&= \max_i \rho_i \sum_j F_{ij} \\
&< \max_i \rho_i && \text{(grâce à la propriété (3))} \\
&< 1
\end{aligned}$$

Nous en déduisons l'existence d'une unique solution au système (1.9). Cette solution est par ailleurs l'unique limite de l'itération linéaire de Jacobi :

$$B^{(n+1)} = E + MB^{(n)} \quad (1.13)$$

Nous avons donc également :

$$B = \left(\sum_{k=0}^{\infty} M^k \right) E$$

Le système linéaire (1.9) donnant les valeurs de radiosité B_i sur les surfaces élémentaires est un système *plein* au sens où relativement peu de ses coefficients sont nuls. Pour une scène de taille moyenne, le nombre N de ses inconnues dépasse facilement plusieurs dizaines de milliers, ce qui rend totalement inapplicables des méthodes directes de résolution de systèmes linéaires, dont le coût est en $O(N^3)$ ³.

D'autre part, du fait que l'on se satisfait souvent d'une solution approximative, les méthodes itératives (méthode de *Jacobi*, de *Gauss-Seidel*) dont le coût d'une itération est $O(N^2)$, paraissent plus indiquées. La relation $S_\infty(M) < 1$ assure la convergence des méthodes itératives de *Jacobi* et *Gauss-Seidel* pour résoudre le système (1.12). En pratique, la méthode de *Gauss-Seidel* est préférée à celle de *Jacobi* parce qu'elle répartit plus rapidement l'énergie dans la scène au cours des premières itérations, du fait qu'elle prend en compte les valeurs calculées pour toutes les lignes précédentes à l'itération courante.

La méthode de *Gauss-Seidel* admet une interprétation physique simple : pour chaque ligne du système on effectue successivement $B_i = E_i + \rho_i \sum_j F_{ij} B_j$. Cela signifie que pour chaque surface A_i on ajoute à B_i la contribution de la radiosité de toutes les autres surfaces A_j .

Une autre méthode itérative, la méthode de *Southwell* [SP94, GC93] est également utilisée, et admet une interprétation physique "duale" : pour chaque surface A_i , on ajoute à toutes les autres surfaces A_j la contribution de A_i à la radiosité B_j . Cette seconde méthode a l'avantage d'illuminer très rapidement la scène au cours des premières itérations, si l'on choisit de commencer par envoyer de l'énergie depuis les sources de lumière.

Le point de départ $B^{(0)}$ des itérations linéaires n'influent pas sur la limite de l'itération, le vecteur initial des valeurs de radiosité est souvent choisi nul (ou égal au vecteur des émittance). Un choix judicieux de $B^{(0)}$, anticipant la répartition générale de la lumière dans la scène, permet cependant d'obtenir plus rapidement une image correcte [CCWG88].

1.2 Méthodes hiérarchiques de Radiosité

L'utilisation de méthodes hiérarchiques dans le cadre de la radiosité provient de l'analogie avec les problèmes d'interaction à N corps, et fut introduite par Hanrahan [HSA91]. Nous décrivons dans cette section le principe de base de la méthode de radiosité hiérarchique, ainsi que celui d'une autre méthode hiérarchique plus générale : le *clustering*.

³Dans le cas de la méthode *Gauss*, son instabilité suffit à la rendre inutilisable pour $N > 100$

1.2.1 Principe

Dans des scènes de complexité moyenne (1000 polygones et plus), la méthode de radiosité est rapidement saturée par la finesse du maillage nécessaire pour reproduire les ombres des objets, ou simplement prendre en compte les variations de l'éclairage sur des polygones de grande taille. Cela vient du fait que, dans la méthode de radiosité, le maillage est affiné indépendamment pour les contributions de toutes les surfaces à la radiosité d'un élément. Par exemple, lorsque deux surfaces A_i et A_j sont très proches il peut être nécessaire de subdiviser A_i pour y calculer la contribution de A_j avec suffisamment de précision (C.f figure 1.8). En revanche, si l'on calcule la contribution de A_i à la radiosité d'une surface A_k située très loin, on s'aperçoit que la corrélation entre le noyau de la radiosité et la radiosité sur A_i est faible :

$$\begin{aligned} b(x) &= e(x) + \rho(x) \int_{A_i} b(y) G(x,y) dy \\ \# \quad & e(x) + \rho(x) \int_{A_i} b(y) dy \int_{A_i} G(x,y) dy \\ \# \quad & e(x) + \rho(x) F_{x,A_i} \int_{A_i} b(y) dy \end{aligned}$$

Cela signifie que, dans ce cas, la radiosité moyenne sur A_i suffit à exprimer correctement le transfert, et donc que l'interaction $A_i \rightarrow A_k$ peut être calculée en considérant A_i comme une seule surface élémentaire dont la radiosité est la moyenne des radiosités de ses sous-surfaces.

De même, il peut se produire que A_i n'ait pas besoin d'être subdivisée pour calculer la lumière reçue depuis A_j mais doit l'être pour celle qu'elle envoie à A_k .

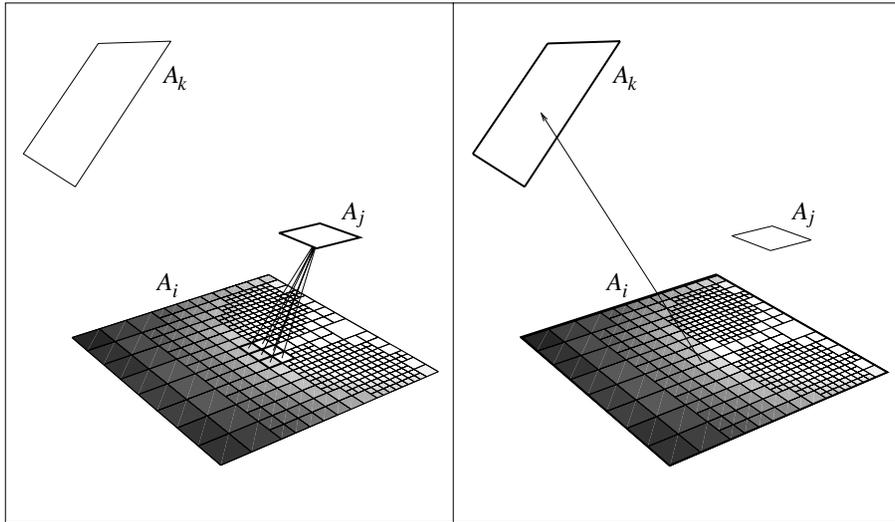


FIG. 1.8: Exemple de surface subdivisée pour la réception uniquement

Le principe de base de la radiosité hiérarchique est d'établir les transferts d'énergie au niveau le plus adéquat relativement à la précision voulue et au temps de calcul autorisé. Les interactions énergétiques sont donc établies au niveau le plus haut possible, pour économiser un maximum de calculs mais suffisamment bas pour contenir l'erreur sous un seuil fixé par l'utilisateur.

Pour cela, les interactions entre surfaces sont symbolisées par des *liens*. Le niveau hiérarchique d'établissement des liens dépendant de la radiosité sur les surfaces, les liens doivent éventuellement être raffinés selon les valeurs de radiosité nouvellement calculées, pour respecter le seuil de précision imposé par l'utilisateur. D'autre part, comme les liens

propagant de l'énergie lumineuse à différents niveaux dans la hiérarchie, la cohérence de la représentation multi-échelles de la radiosité doit être assurée après chaque transfert d'énergie. Chaque itération de radiosité hiérarchique est donc généralement composée de trois phases : le *raffinement* (ou *Refine*), la propagation de l'énergie le long des liens (ou *Gather*) et la distribution des valeurs de radiosité sur tous les niveaux hiérarchiques (ou *Push/Pull*). Nous allons maintenant expliquer plus en détail ces trois étapes élémentaires.

(a) Raffinement

La phase de raffinement consiste à parcourir l'ensemble des liens, et pour chacun d'eux, décider si l'approximation du transfert d'énergie qu'il représente est suffisamment précise. Si oui, le lien est laissé tel quel ; sinon il est *raffiné*, ce qui signifie que l'on subdivise l'émetteur ou le récepteur, créant ainsi autant de nouveaux liens (C.f figure 1.9) qui sont à leur tour examinés récursivement. Le raffinement d'un lien s'arrête lorsque la précision du transfert pour ce lien est atteinte, ou lorsque l'aire des surfaces qu'il relie passe en dessous d'un seuil limite, également fixé par l'utilisateur.

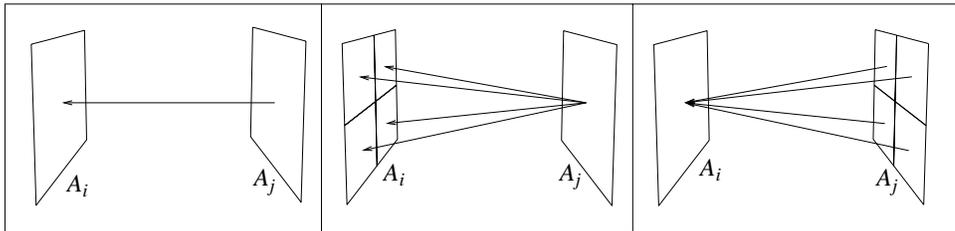


FIG. 1.9: Un lien représentant un transfert d'énergie d'une surface A_j vers une surface A_i (à gauche) a été raffiné, au centre, par raffinement du récepteur, et à droite, par raffinement de l'émetteur.

Le processus de raffinement nécessite donc de pouvoir estimer la qualité de l'approximation réalisée en calculant le transfert d'énergie entre deux surfaces par un unique facteur de forme. L'erreur d'approximation est en général estimée à l'aide d'un *oracle de raffinement*, prenant en compte différents aspects de l'approximation : variations de la radiosité sur l'émetteur, présence d'obstacles entre l'émetteur et le récepteur, quantité d'énergie transférée, qualité de l'approximation du facteur de forme lui-même. L'oracle de raffinement sert également à choisir, si nécessaire, entre la subdivision de la source ou du récepteur. Nous présenterons au chapitre 2 (section 2.3.3) et au chapitre 8 des exemples de tels oracles de raffinement.

Les surfaces élémentaires utilisées en radiosité hiérarchiques sont habituellement des triangles et des quadrilatères. Elles sont généralement subdivisées comme l'indique la figure 1.10.

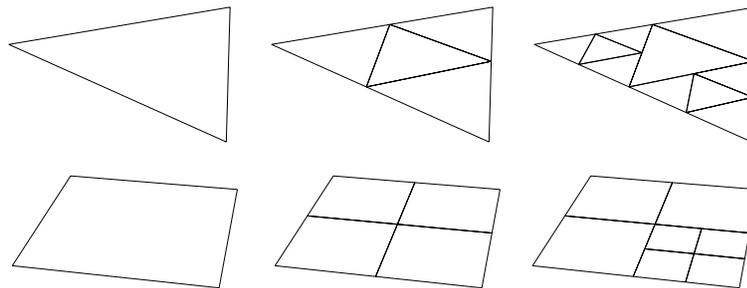


FIG. 1.10: Subdivision des quadrilatères et des triangles

(b) Propagation de l'énergie

La seconde phase de l'itération sert à propager l'énergie le long des liens établis par l'opération de raffinement. L'ensemble des liens est parcouru ; pour chacun d'entre eux on calcule le transfert d'énergie correspondant à partir de la radiosité B_j de la surface émettrice et du facteur de forme F_{ij} du récepteur A_i vers l'émetteur A_j :

$$I_{ij} = F_{ij}B_j$$

La valeur obtenue est stockée au niveau du récepteur. Elle ne représente pas de la radiosité, mais de l'*irradiance*, que l'on convertit en radiosité par produit avec la réflectance du récepteur lors de la phase suivante.

(c) Push/Pull

Comme nous l'avons dit, la phase de *Push/Pull* sert à assurer la cohérence des valeurs de radiosité issues de la propagation de l'énergie effectuée lors de la phase précédente. La radiosité sur les surfaces émettrices doit en effet être mise à jour en tenant compte de l'énergie apportée à tous les niveaux de la hiérarchie. Ceci est généralement effectué par un parcours en profondeur de la hiérarchie (C.f figure 1.11) : lors de la descente, les valeurs d'irradiance sont ajoutées jusqu'au niveau des feuilles de la hiérarchie, où leur valeur cumulée est multipliée par la valeur de réflectance de la feuille, pour obtenir la radiosité de cette feuille. Lors de la remontée, la radiosité d'un élément est calculée comme la moyenne des radiosités de ses fils pondérées par les aires des sous-surfaces associées.

A l'issue de la phase de *Push/Pull*, la radiosité est représentée de manière cohérente à tous les niveaux hiérarchiques du maillage.

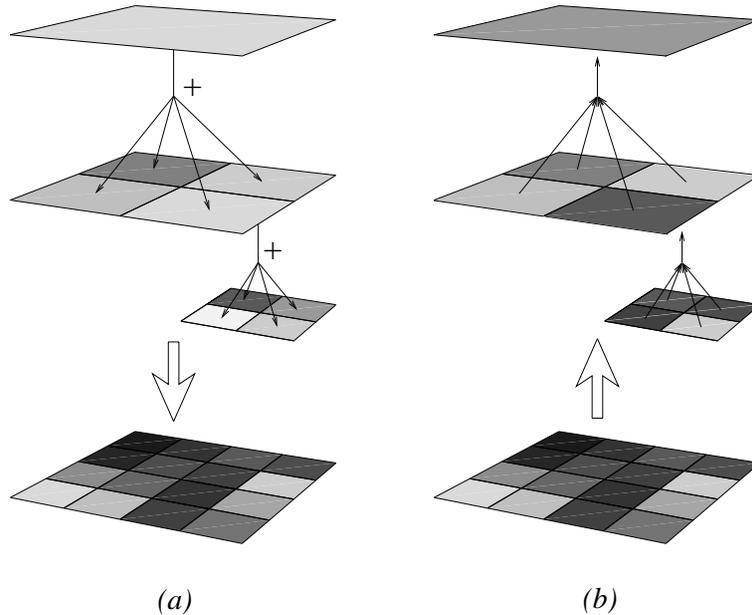


FIG. 1.11: Illustration de la phase de *Push/Pull* permettant de rendre cohérente la représentation hiérarchique de la radiosité. (a) Au cours de la descente, les valeurs d'irradiance sont accumulées jusqu'au niveau des feuilles. (b) Les valeurs de radiosité sont moyennées lors de la remontée.

La méthode de radiosité hiérarchique consiste à enchaîner les trois phases que nous venons de décrire, autant de fois que nécessaire pour obtenir une approximation raisonnable

de la solution de radiosité cherchée. L'algorithme est initialisé en établissant un lien entre chaque paire de surfaces constituant la scène originale.

Le fait d'établir des transferts d'énergie à un niveau le plus haut possible économise un nombre important d'interactions et donc de calculs de facteurs de forme. En pratique, on mesure que le nombre de liens créés est de l'ordre de $s^2 + p$, où s est le nombre de surfaces originales constituant la scène et p le nombre total de sous-surfaces.

1.2.2 Généralisation de la hiérarchie : le *clustering*

Malgré le gain d'efficacité qu'elle représente par rapport à la méthode de radiosité, la radiosité hiérarchique telle que nous venons de la présenter possède un inconvénient majeur : son coût reste en $O(s^2)$ pour une scène initialement constituée de s surfaces, puisqu'au départ tous les couples de surfaces constituant la scène doivent être examinés. Ce coût est particulièrement limitatif de la méthode de radiosité hiérarchique pour des scènes contenant des objets complexes, constitués de beaucoup de petits polygones.

Pour y remédier, il est possible de généraliser la hiérarchie introduite précédemment en regroupant à leur tour les surfaces initiales en groupements appelés *clusters*, puis les *clusters* entre eux en *clusters* plus gros, selon des critères de proximité et de taille, pour obtenir une hiérarchie complète dont la racine est un unique *cluster* contenant toute la scène, et les feuilles sont les surfaces élémentaires sur lesquelles la radiosité est représentée par les valeurs B_i . La figure 1.12 montre des exemples de telles hiérarchies.

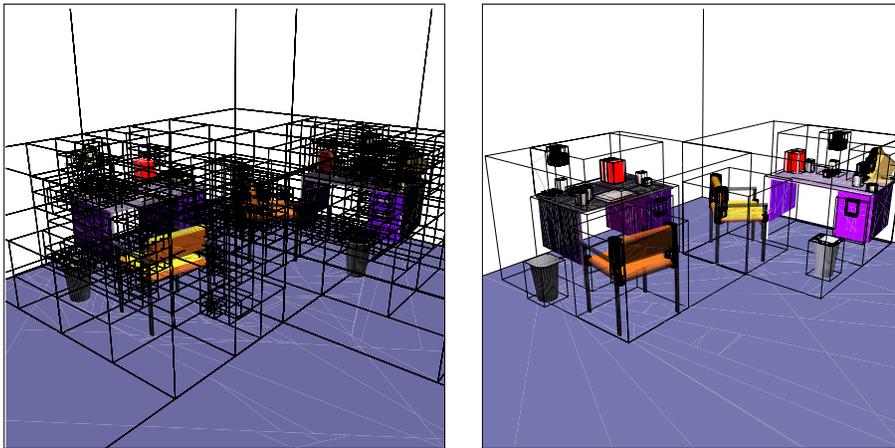


FIG. 1.12: Exemple de hiérarchies de clusters. A gauche : *Octree*. A droite : *boîtes englobantes (H.B.V)*. Les polygones constituant le chapeau de la lampe sont groupés dans un cluster qui fait lui-même partie d'un cluster plus gros contenant toute la lampe, etc.

La construction même d'une hiérarchie de *clusters* est un problème difficile si l'on veut respecter une certaine cohérence dans les groupements réalisés [CDP95] : dans l'exemple ci-dessus, il serait par exemple peu judicieux de grouper une des facettes de la poubelle avec un pied de la chaise d'à côté dans un premier *cluster*, et les autres facettes dans un autre.

Plusieurs types de hiérarchies de *clusters* permettent d'effectuer des calculs de radiosités : les *Hierarchical Bounding Volumes* (ou H.B.V) sont des boîtes minimales aux cotés parallèles aux plans définis par les axes du repère de coordonnées [GS87]. Les *Octrees* sont des boîtes obtenues par subdivision récursive d'un parallélépipède rectangle en huit parallélépipèdes de dimensions deux fois moindres [Gla84].

Contrairement à la méthode de radiosité hiérarchique, la première itération débute par le raffinement d'un unique lien du *cluster* racine vers lui-même, ce qui a pour effet de réduire considérablement le nombre de liens utilisés par l'algorithme. En pratique on observe que

ce nombre de liens est de l'ordre de $s \log s + p$ [SAG94] où s est le nombre de surfaces initiales de la scène, et p le nombre total de sous-surfaces.

L'utilisation de *clusters* introduit de nouvelles questions, spécifiques au calcul de transferts d'énergie entre éléments non plans :

Les calculs des facteurs de forme *cluster-cluster*, *surface-cluster*, et *cluster-surface* doivent être menés correctement, en prenant éventuellement en compte la *visibilité intra-cluster* qui fait qu'un *cluster* masque en partie la lumière qu'il reçoit ou envoie [SSS97]. Alors qu'en radiosité hiérarchique, les facteurs de forme F_{ii} sont nuls puisque les surfaces sont planes, le facteur de forme d'un *cluster* vers lui-même n'est pas nécessairement nul, et son calcul peut s'avérer coûteux. En considérant qu'un *cluster* reçoit de l'énergie de lui-même ou des autres *clusters*, ce facteur de forme particulier peut être approché par l'expression suivante, impliquant les facteurs de forme des liens arrivant sur le *cluster* [SSS97] :

$$F_{ii} = 1 - \sum_j F_{ij}$$

Il ne s'agit que d'une approximation, car la scène n'est pas nécessairement fermée.

Dans le cas du *clustering*, les algorithmes de raffinement, propagation de l'énergie et *Push/Pull* sont naturellement étendus à toute la hiérarchie des *clusters* et des surfaces.

1.2.3 Expression matricielle

Nous aimerions obtenir une formulation des méthodes hiérarchiques de radiosité, équivalente à l'expression matricielle (1.12) obtenue pour la méthode de radiosité. Notons B le vecteur des valeurs de radiosité sur les feuilles de la hiérarchie, B_E celui des valeurs de radiosité sur les surfaces émettrices (*i.e* celles desquelles part un lien) et B_R celui des valeurs de radiosité sur les surfaces réceptrices (*i.e* celles sur lesquelles aboutit un lien). Nous notons également N la taille de B , n_1 la taille de B_E et n_2 la taille de B_R .

Soit enfin Q la matrice $n_1 \times N$ qui transforme B en B_E . Cette matrice permet de calculer la radiosité sur les surfaces émettrices en fonction des valeurs de radiosité B_i des feuilles de la hiérarchie. Soit P la matrice $N \times n_2$ transformant B_R en B . Cette matrice répartit la radiosité des récepteurs vers les feuilles de la hiérarchie. Soit M la matrice $n_2 \times n_1$ liant B_E à B_R . La matrice M contient donc les facteurs de forme de tous les liens établis dans la hiérarchie.

Nous avons par définition :

$$B_E = QB \quad B = PB_R$$

et donc :

$$B = E + \rho \underbrace{PMQ}_M B$$

avec (Voir aussi la figure 1.13) :

$$Q = \begin{bmatrix} \frac{a_1}{S_1} & \frac{a_2}{S_1} & \frac{a_3}{S_1} & \frac{a_4}{S_1} & 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & \frac{a_5}{S_2} & \dots & \dots & 0 & \dots & 0 \\ \vdots & & & & & & & & & \vdots \\ \frac{a_1}{S_l} & \frac{a_2}{S_l} & \frac{a_3}{S_l} & \frac{a_4}{S_l} & \frac{a_5}{S_l} & \dots & \frac{a_j}{S_l} & 0 & \dots & 0 \\ \vdots & & & & & & & & & \vdots \\ \frac{a_1}{S_{n_1}} & \frac{a_2}{S_{n_1}} & \frac{a_3}{S_{n_1}} & \frac{a_4}{S_{n_1}} & \dots & & & & & \frac{a_N}{S_{n_1}} \end{bmatrix} \leftarrow l^{\text{ème}} \text{ ligne}$$

↑
 $j^{\text{ème}}$ colonne

$$\rho = \begin{bmatrix} \rho_1 & 0 & \dots & 0 \\ 0 & \rho_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \rho_N \end{bmatrix} \quad N$$

$$M = \begin{bmatrix} F_{11} & \dots & F_{1n_1} \\ \vdots & & \vdots \\ F_{n_21} & \dots & F_{n_2n_1} \end{bmatrix} \quad \begin{matrix} n_2 \\ n_1 \end{matrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 1 & & & \\ 1 & 0 & 0 & 0 & 1 & & & \\ 1 & 0 & 0 & 0 & 1 & & & \\ 0 & 1 & 0 & 0 & 1 & & & \\ 0 & 1 & 0 & 0 & 1 & & & \\ 0 & 1 & 0 & 0 & 1 & & & \\ 0 & 0 & 1 & 0 & 1 & & & \\ \vdots & \vdots \end{bmatrix} \quad N$$

Le coefficient d'indices (i, j) de la matrice M vaut donc (C.f figure 1.13) le facteur de forme du lien partant d'un parent S_l de la feuille A_j vers un parent S_k de la feuille A_i , pondérés par le rapport des surfaces de A_j et S_l .

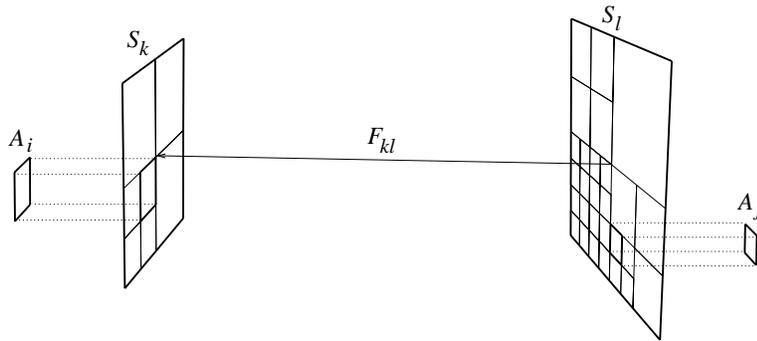


FIG. 1.13: Contribution d'un lien au coefficient (i,j) de la matrice de radiosity hiérarchique M . Étant donné deux feuilles A_i et A_j , il existe au plus un lien propageant de l'énergie d'un parent S_l de A_j vers un parent S_k de A_i .

La quantité $S_\infty(M)$ vaut donc la valeur maximum sur toutes les feuilles A_i , de la somme de ces facteurs de forme, en considérant toutes les surfaces A_j participant à l'illumination sur A_i .

1.2.4 Convergence

Prouver la convergence de la méthode de radiosity hiérarchique n'est pas aussi simple que pour la radiosity classique. On peut en effet montrer qu'il existe des configurations pour lesquelles $S_\infty(M) > 1$, comme illustré sur la figure 1.14.

Notons en effet B_α la radiosity sur un élément α (α remplaçant a, b, c ou d). On a, dans la configuration présentée sur la figure 1.14 :

$$B_d = F_{c \cup d, a} B_a + F_{d, b} B_b$$

Et donc, dans ce cas précis :

$$S_\infty(M) > F_{c \cup d, a} + F_{d, b} \quad \# \quad 1.48755623$$

Cela signifie qu'un élément peut recevoir plus d'énergie qu'il n'y en a dans la scène.

En inversant le sens de tous les liens représentés, on obtient une configuration valide dans laquelle la somme des coefficients de la matrice M dans la colonne correspondant à la contribution de d aux autres éléments est également supérieure à 1 (Ici exactement 1.4875562). Il existe donc des situations dans lesquelles un élément envoie plus d'énergie qu'il n'en possède.

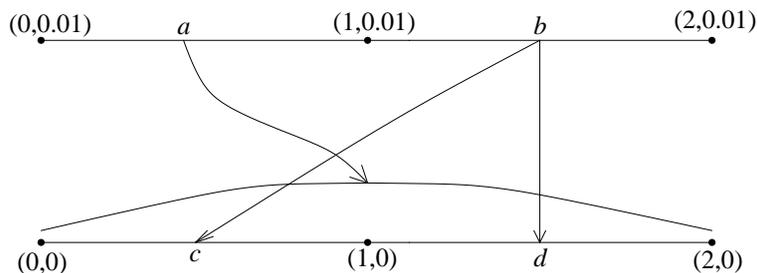


FIG. 1.14: Exemple de situation dans laquelle la somme des facteurs de forme associés aux contributions des différents éléments à la radiosité d'un élément particulier (ici, l'élément d) est supérieure à 1 (Voir texte).

Nous n'avons cependant pas prouvé par ce raisonnement que dans de tels cas, l'itération diverge. D'ailleurs, dans le cas précis de la figure 1.14, on a $\rho(M) \approx 0.99$. Nous verrons cependant à la section 2.3 comment effectuer l'itération de manière à assurer que le rayon spectral de la matrice effectivement utilisée reste inférieur à l'unité, bien qu'il semble que cette technique ne soit pas nécessaire dans ce cas précis.

1.2.5 Rendu de la solution

Une fois la radiosité calculée à tous les étages de la hiérarchie, la solution peut être rendue à l'écran. Le fait de traduire une valeur de radiosité en une couleur de pixels à l'écran constitue en lui-même un problème délicat [SP94]. Nous en parlerons à la section 2.1.5 dans le cadre du recensement des approximations dans la méthode de radiosité hiérarchique.

Bien que la méthode de radiosité hiérarchique, telle que nous l'avons présentée, calcule des valeurs uniformes de radiosité sur des surfaces élémentaires, diverses techniques permettent d'éviter un aspect crénelé des images rendues. Nous présentons brièvement deux telles techniques : l'*interpolation de Gouraud* (ou *smooth shading*) et la *récolte finale* (ou *final gather*).

(a) L'interpolation de Gouraud

La technique l'interpolation de Gouraud s'applique à un triangle connaissant la couleur de ses sommets, et consiste à rendre chaque point du triangle avec, comme couleur, la moyenne des couleurs des sommets, pondérées par les coordonnées barycentriques du point rendu. Le triangle est alors dessiné avec une variation progressive de la couleur. Pour appliquer cette technique à des polygones à plus de trois sommets, on les subdivise en triangles. Dans la pratique les bibliothèques graphiques telles que **OpenGL** effectuent ce découpage elles-mêmes en utilisant les valeurs de couleurs au sommet du polygone original.

Dans la méthode de radiosité hiérarchique, il faut donc calculer des valeurs de couleurs aux sommets des polygones en chaque feuille de la hiérarchie, *i.e* à chaque nœud du maillage des surfaces. Ce calcul est effectué au moment du *push/pull* : on ajoute la radiosité de chaque feuille à chacun des nœuds du maillage correspondant aux sommets du polygone, pondérée par les poids indiqués sur la figure 1.15.

Afin d'avoir un meilleur contrôle du découpage des polygones en triangles, il est préférable d'effectuer ce découpage soi-même. On commence pour cela par imposer une contrainte de subdivision sur le maillage : deux mailles voisines ne doivent pas avoir plus d'un étage de subdivision d'écart. On triangule ensuite toutes les mailles à plus de quatre sommets. La figure 1.16 illustre ce procédé sur un maillage rectangulaire.

En procédant ainsi, on assure une parfaite continuité de la radiosité dans les surfaces rendues à l'écran.

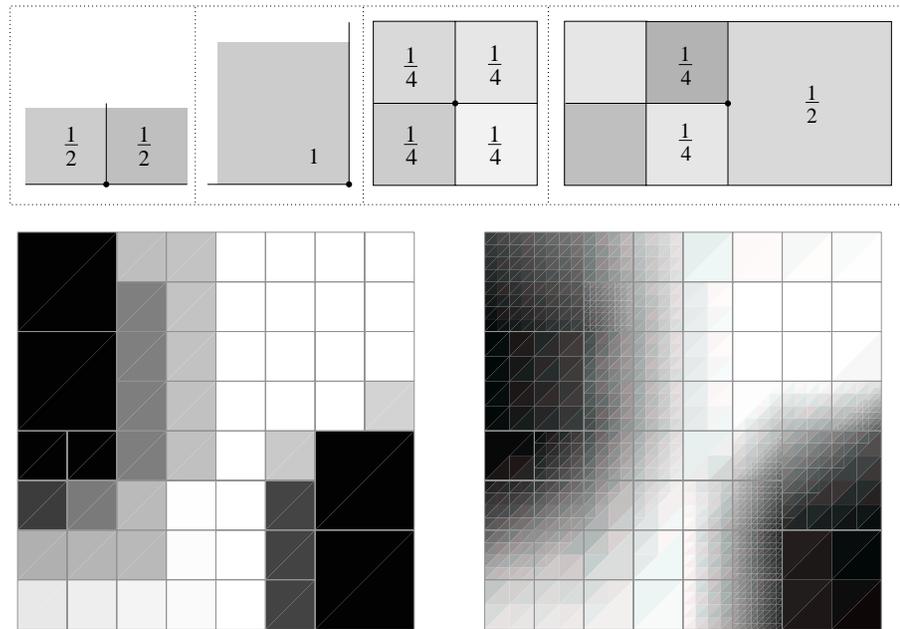


FIG. 1.15: En haut : *calcul des valeurs aux sommets du maillage en vue de l'interpolation de la radiosité sur chaque face (smooth-shading)*. En bas : *effet produit par l'interpolation*.

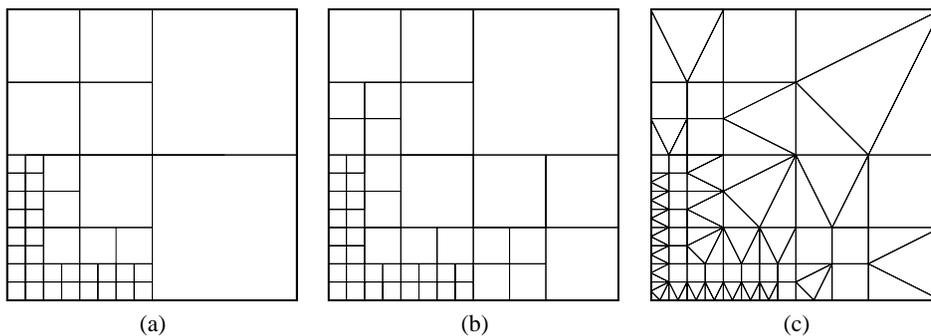


FIG. 1.16: *Exemples de quadtree (a) libre (b) contraint (c) triangulé*.

(b) Récolte finale

La *récolte finale* (ou *final gather*) est une opération consistant à évaluer la couleur de chaque point x de la scène rendue à l'écran (la contribution d'un point à l'image rendue peut être déterminée par lancer de rayons [GAC⁺88]) en effectuant un calcul précis de sa radiosité, exprimée en fonction de la radiosité calculée dans la scène :

$$b(x) = \rho(x) \sum_i B_i F_{x,A_i}$$

Le facteur de forme F_{x,A_i} peut-être calculé par intégration numérique, ou pour gagner du temps, remplacé par une valeur représentative du noyau multiplié par la visibilité. Étant donné que seuls les points de la scène visibles à l'écran sont concernés par ce calcul, cette technique doit être appliquée de nouveau lorsque l'on change de point de vue.

Chapitre 2

Contrôle de l'erreur en radiosité hiérarchique

EN TANT QUE méthodes numériques de simulation de l'éclairage, les méthodes hiérarchiques de radiosité que nous avons présentées au chapitre 1 sont basées sur un grand nombre d'approximations. Certaines de ces approximations sont implicitement contenues dans la modélisation du problème ou dans les limitations du matériel informatique utilisé, et sont par là indépendantes de la volonté de l'utilisateur. D'autres en revanche sont basées sur un calcul arbitrairement approximatif de certaines quantités, comme les facteurs de forme, dans le but d'accélérer le calcul des solutions au détriment de la précision du résultat. De telles approximations sont par exemples utilisées pour obtenir une solution grossière, soit par manque de temps, soit pour régler des paramètres de scène. Dans le cadre de la simulation numérique, elles n'ont d'intérêt que si l'on sait quantifier leur impact sur l'erreur totale, de façon à pouvoir garder un contrôle sur la précision de la solution calculée.

Dans cette optique, et après avoir passé en revue les différentes sources d'approximations dans la méthode de radiosité hiérarchique (section 2.1), nous nous intéresserons plus particulièrement à l'impact des approximations des facteurs de forme sur la solution de l'équation discrète de radiosité, puisque les différentes approximations de la visibilité peuvent être traduites sous cette forme.

L'étude de la relation entre l'erreur sur les facteurs de forme et l'erreur sur la solution du système nous permettra de conclure que le contrôle de l'erreur due aux approximations de visibilité peut être réalisé grâce à un algorithme de calcul contrôlé des facteurs de forme. Un tel algorithme, basé sur le dosage des approximations de la visibilité sera alors présenté à la partie II.

2.1 Caractérisation de l'erreur

Comme pour toute méthode numérique de résolution d'un problème physique, De nombreuses approximations interviennent dans les méthodes hiérarchiques de radiosité, au cours du processus conduisant du phénomène observé à une simulation numérique [ATS94]. Ces approximations apparaissent à différents niveaux du processus, et peuvent être classées, par ordre chronologique, dans les catégories suivantes : approximations de modélisation, approximations de discrétisation, approximations de résolution du problème discret, et enfin, approximations de visualisation de la solution calculée. Nous allons rapidement examiner les différentes sources d'erreur dans la méthode de radiosité hiérarchique en suivant ce classement.

2.1.1 Modélisation

Plusieurs hypothèses sont utilisées dans la modélisation des phénomènes de propagation de la lumière pour aboutir à l'équation d'équilibre des échanges lumineux (1.8) introduite au chapitre 1. Elles concernent à la fois la géométrie de la scène, la modélisation de l'énergie lumineuse et les propriétés d'interaction de la lumière avec l'environnement.

Approximation de la géométrie

Dans l'optique d'une représentation informatique, la géométrie du monde réel est approchée par des courbes polynomiales [Sab85, SB86, NN93] ou des surfaces de subdivision [DKT98, Sta98]. Il s'agit en effet de pouvoir accéder efficacement aux coordonnées des points des surfaces modélisées et d'effectuer rapidement les calculs d'intersection de la géométrie avec les droites de l'espace symbolisant les rayons lumineux, en vue de déterminer la visibilité.

A un niveau microscopique ces approximations reviennent à simplifier considérablement la propagation de la lumière à la surface des matériaux, ce qui a principalement pour effet de donner aux surfaces l'"aspect synthétique" caractéristique des images de synthèse. Au niveau macroscopique, les approximations géométriques transforment les objets en surfaces visuellement plus simples, d'aspect généralement polygonal, ou possédant des courbures inappropriées.

Modélisation de la lumière

Comme pour de nombreuses applications, la simulation de l'éclairage utilise la formulation de l'optique géométrique, dans laquelle la lumière se déplace en ligne droite, et les rayons lumineux se trouvent donc modélisés par des droites de l'espace. Les phénomènes d'interférence sont la plupart du temps négligés, parce qu'ils requièrent un traitement particulier [SM92].

Enfin, le spectre continu de longueurs d'ondes caractérisant l'énergie lumineuse est habituellement modélisé par un triplet de longueurs d'ondes, qui sont celles des trois couleurs de base servant à l'affichage des écrans d'ordinateurs : le rouge, le vert et le bleu. Ces couleurs de base étant également celles perçues par l'œil humain, cette approximation nous est pratiquement insensible. Représenter un spectre contenant d'autres longueurs d'ondes demande d'effectuer une substitution adéquate par un triplet de valeurs R, V, B .

Interactions avec l'environnement

La lumière interagit avec l'air ambiant sous la forme d'effets de diffusion et de réfraction. Des techniques particulières ont été développées pour simuler ces phénomènes dans l'atmosphère sous différentes conditions [JC98, NND96, Nis98], également dans le cadre de méthodes hiérarchiques de radiosité [AH93, Sil95], pour des scènes impliquant des *milieux participants* [PPS97]. Les phénomènes de diffusion sont cependant généralement négligés dans la méthode de radiosité hiérarchique, notamment pour son utilisation dans des scènes d'intérieur, dont l'échelle ne rend pas significatif l'impact d'effets de ce type. On néglige également les phénomènes de diffraction, car ils sont qualitativement de peu d'importance. Quant aux phénomènes de réfraction, ils nécessitent, par leur caractère non linéaire et non indépendant de la longueur d'onde, un traitement spécial [MH92, Wat90].

L'interaction de la lumière avec des surfaces est modélisée par une fonction de réflectance bidirectionnelle (voir la section 1.1.1.(d)), traduisant la proportion de la lumière qui, incidente selon une certaine direction, est réfléchie dans une autre direction. Comme nous l'avons vu au cours du chapitre 1, de nombreuses hypothèses de régularité et de symétrie sont utilisées pour simplifier cette représentation.

Le modèle de réflexion spéculaire pure est principalement utilisé en lancer de rayon [BDM84]. Bien que des méthodes aient été développées pour générer [HTSG91] et traiter des modèles

de réflexion arbitraire [SAWG91], l'apanage de la méthode de radiosité hiérarchique est l'hypothèse de réflexion idéalement diffuse, selon laquelle la lumière est réfléchie uniformément dans toutes les directions du demi-espace tangent à la surface. Certains modèles de réflexion plus complexes, comme le modèle de *Phong* [Pho75], sont également implémentés dans les bibliothèques graphiques et souvent directement disponibles dans le matériel, mais pas utilisés dans l'algorithme de radiosité hiérarchique classique.

Une dernière hypothèse concerne l'indépendance des longueurs d'onde lors de la réflexion sur une surface : chaque composante est traitée séparément et ne modifie pas les autres. Il s'agit là encore d'une hypothèse de linéarité très souvent utilisée.

Disposant de propriétés et d'une géométrie simplifiées, on établit alors une équation d'équilibre reliant les différentes quantités. Cette équation n'est donc plus celle vérifiée par l'énergie lumineuse, mais en constitue une approximation utilisable dans la plupart des applications [Kaj86].

2.1.2 Discrétisation

Une fois modélisé par une équation continue, l'équilibre de l'énergie lumineuse dans une scène est résolu de manière approchée en cherchant sa solution dans un espace vectoriel de fonctions simples. Des approches utilisant des fonctions polynomiales [Zat93] ou des ondelettes [GSCH93] ont été développées, certaines prenant en compte l'erreur de discrétisation dans l'oracle de raffinement du maillage [BW96]. L'utilisation de fonctions constantes par morceaux a cependant conservé son succès à cause de sa simplicité d'utilisation et de son efficacité. Grâce au *smooth shading*, dont nous avons parlé à la section 1.2.5.(a), le rendu peut en effet s'effectuer sans les discontinuités d'illumination qui caractérisent ce type d'approximation.

Bien que cela ne soit pas nécessaire, la discrétisation de l'équation d'équilibre s'accompagne souvent d'une discrétisation de la géométrie [BMSW91], notamment utile au moment du rendu de la solution.

La solution obtenue appartient donc à un espace restreint de fonctions qui ne vérifient pas toujours les propriétés de la distribution de la radiosité (dérivabilité, vitesse de variation) et vérifie une équation qui n'est pas l'équation originale d'équilibre de l'énergie lumineuse, mais un système linéaire.

2.1.3 Calcul des coefficients du système linéaire

Le calcul des coefficients du système linéaire intervenant dans les méthodes de radiosité, représente à lui seul une importante difficulté, puisqu'on ne connaît d'expression analytique informatiquement utilisable¹ du facteur de forme entre deux polygones que dans les cas de visibilité totale entre ces polygones. Les facteurs de forme sont donc estimés par le biais d'approximations diverses :

Une approximation très communément utilisée consiste à négliger la corrélation entre le noyau de l'équation de radiosité et le terme de visibilité, ce qui permet d'intégrer séparément les deux termes correspondants. Cette approximation est utilisée dans deux formulations différentes du facteur de forme entre deux surfaces A_i et A_j :

En utilisant la formule (1.10) pour calculer le facteur de forme F_{ij} , on calcule le facteur

¹C'est à dire sans intégrales, ni séries, ni fonctions complexes.

de forme point-polygone F_{x,A_j} en négligeant la corrélation noyau-visibilité :

$$F_{x,A_j} = \int_{A_j} k(x,y)v(x,y)dy$$

$$\# \underbrace{\int_{A_j} k(x,y)dy \int_{A_j} v(x,y)dy}_{\bar{F}_{x,A_j}}$$

On calcule ensuite le facteur de forme F_{ij} comme :

$$F_{A_i,A_j} = \frac{1}{a_i} \int_{A_i} \left(\int_{A_j} k(x,y)dy \int_{A_j} v(x,y) \right) dx$$

L'erreur apparaît donc comme la moyenne du terme $\varepsilon(x) = \bar{F}_{x,A_j} - F_{x,A_j}$ sur la surface A_j . Le signe de ε dépend du rapport des distances du point x aux parties visibles et masquées de la source vue depuis x . Pour des obstacles complexes la fonction $x \mapsto \varepsilon(x)$ change rapidement de signe sur le récepteur, et sa moyenne est très faible, ce qui justifie l'approximation proposée. Il existe cependant des situations dans lesquelles ε est de signe quasiment constant, comme le montre la figure 2.1. Dans ce cas, l'erreur d'approximation sur le facteur de forme peut atteindre 10%, dans des configurations bien choisies. L'erreur ε admet des variations très lentes, de classe C^∞ , et une localisation subtilement dissimulée dans les zones de pénombre, passe en revanche totalement inaperçue au seul regard de l'œil humain, qui est beaucoup plus apte à détecter des discontinuités de l'éclairage ou de ses dérivées.

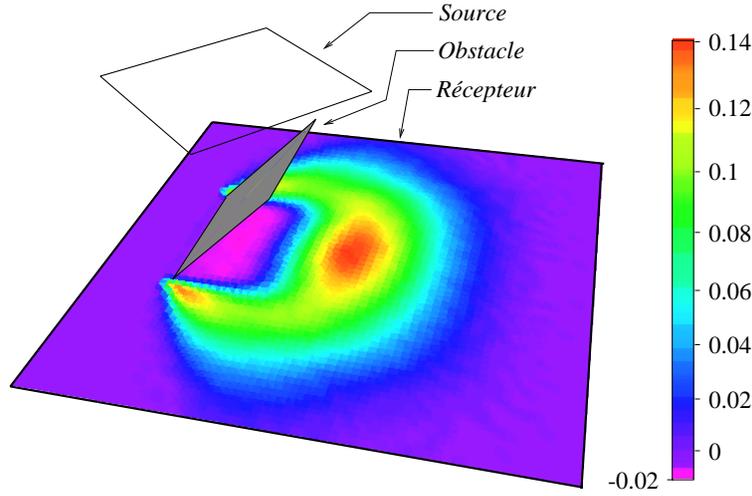


FIG. 2.1: Erreur de corrélation noyau-visibilité dans le calcul du facteur de forme point-polygone dans une configuration particulièrement défavorable : pour de nombreux points x du récepteur, la partie masquée de la source correspond aux points y pour lesquels $k(x,y)$ est minimal. Seuls font exception les points situés juste sous l'obstacle, pour lesquels la situation s'inverse. L'erreur moyenne est donc positive et ici de l'ordre de 2.55%.

Une autre approximation couramment utilisée pour le facteur de forme est :

$$F_{ij} = \frac{1}{a_i} \int_{A_i} \int_{A_j} k(x,y)v(x,y)dx dy$$

$$\# \underbrace{\frac{1}{a_i} \int_{A_i} \int_{A_j} k(x,y)dx dy}_{F_0} \underbrace{\int_{A_i} \int_{A_j} v(x,y)dx dy}_{\bar{v}}$$

La quantité F_0 désigne alors le facteur de forme *sans occlusion* entre A_i et A_j , que l'on pondère par un terme de visibilité moyenne \bar{v} entre les deux surfaces. Il peut être également calculé grâce à une formule explicite [SH93] (En revanche, le terme \bar{v} est généralement calculé par échantillonnage). L'erreur associée à cette approximation est généralement négligeable, sauf dans des cas similaires à celui décrit sur la figure 2.1.

Enfin, le calcul de F_{x,A_j} est souvent effectué à l'aide d'une discrétisation des directions comme par exemple dans le cas de la technique de l'hémicube (voir la section 1.1.4.(a)). L'effet flagrant de cette discrétisation est l'apparition d'artefacts visuels à grande échelle sur les récepteurs de grande taille [SP94], lorsque la résolution de l'échantillonnage des directions est insuffisante.

2.1.4 Résolution du système

Le système linéaire fournissant la solution approchée du problème de l'équilibre des échanges lumineux est traditionnellement un système "plein", *i.e* dont presque tous les coefficients sont non nuls. Sachant que l'ordre d'un tel système avoisine facilement $n = 10^4$ inconnues, il est hors de question de le résoudre par une méthode directe, dont font partie la *méthode de Gauss*, ou la *méthode QR*, et dont le coût est toujours en $O(n^3)$.

On résout de préférence le système grâce à une méthode itérative telle la *méthode de Gauss-Seidel* ou encore de *Southwell* [GC93]. De telles méthodes n'aboutissent, au bout d'un nombre fini d'itérations qu'à une solution approchée du système. On peut néanmoins établir des bornes sur la distance de l'itéré courant à la solution.

2.1.5 Visualisation de la solution

La dynamique de sensibilité de l'œil humain à la luminosité est d'environ 10^{10} , alors que celle de l'affichage des couleurs sur un écran d'ordinateur dépasse à peine 10^2 . Il est donc impossible de rendre à l'écran la totalité de l'information calculée au cours d'une simulation de l'éclairage. Les couleurs calculées sont donc transformées avant d'être affichées, avec une certaine perte d'information.

Une transformation linéaire des valeurs de luminance permet de conserver le contraste dans les images [War94]. Il est également possible de paramétrer la transformation de manière à faire correspondre le plus petit écart de luminosité perceptible à l'œil à celui représentable à l'écran. Le facteur d'échelle entre ces deux quantités peut être déterminé expérimentalement [CIE81].

2.1.6 Conclusion

Nous avons vu qu'un bon nombre d'approximations interviennent au cours d'une simulation de l'éclairage par la méthode de radiosité hiérarchique.

Ces différentes approximations n'ont pas la même portée en termes d'altération des images. Nous pouvons dégager la tendance suivante : les approximations basées sur la discrétisation, par exemple des directions d'échantillonnage ou des fonctions d'illumination, ont une portée globale restreinte, d'autant qu'elles diminuent avec la finesse de la discrétisation. En revanche elles sont la source d'artefacts visuels qui n'échappent pas à l'œil humain : discontinuités d'illumination ou de ses dérivées entre les surfaces élémentaires, marches d'escalier dans les zones de pénombre.

A l'opposé, les approximations de la visibilité, du noyau du facteur de forme, forment une composante plus "diffuse" de l'erreur qui peut passer inaperçue, mais qui n'échappe pas à un test basé sur une distance mathématique à la solution exacte : par exemple pour une scène globalement sous-éclairée, l'erreur mesurée en utilisant une distance du type $\|B_{calculé} - B_{exact}\|_{L_1}$ sera importante, mais peu visible. A l'inverse, une telle distance ne permet pas de détecter les discontinuités, les ombres inappropriées (pénombre à la place de l'ombre par exemple), ou les valeurs localement erronées [SD95].

La méthode de mesure de l'erreur doit donc être adaptée au type d'application pour laquelle nous effectuons le calcul. Une mesure globale de l'erreur est plus adaptée à estimer l'erreur sur l'équilibre global des échanges lumineux ; pour prendre en compte des artefacts visuels précis, il faut en revanche une mesure qui les traite spécifiquement (ce que nous ferons au chapitre 7), par exemple en contrôlant le phénomène qui les cause.

2.2 Impact de l'approximation des facteurs de forme

Dans cette section, nous nous intéressons plus particulièrement à l'impact des approximations des facteurs de forme sur la précision d'une simulation de l'éclairage, dans le cadre d'une formulation de la radiosité hiérarchique utilisant des fonctions constantes par morceaux.

On distingue généralement deux philosophies d'estimation de ce type d'erreur : lorsque l'on prend en compte la valeur déjà calculée de la radiosité, il s'agit d'une estimation *a posteriori* de l'erreur. En revanche, lorsque l'erreur sur la solution est estimée uniquement en fonction des paramètres d'entrée du système, *i.e* dans notre cas de l'erreur sur les facteurs de forme, il s'agit d'une estimation *a priori* de l'erreur.

Nous présentons dans les sections suivantes deux méthodes d'estimation de l'erreur sur la solution de radiosité à partir de bornes d'erreur sur les facteurs de forme. La première, est une estimation *a posteriori*, inspirée d'une méthode d'estimation d'erreur introduite par Lischinski *et al.*, en 1994 [LSG94] et basée sur la propagation de bornes sur la solution de radiosité. La seconde est une estimation *a priori* de l'erreur, connaissant l'erreur d'approximation des facteurs de forme, que nous avons établie à partir d'outils d'étude de la perturbation des systèmes linéaires.

Nous allons voir que selon l'approche choisie, le lien de cause à effet établi peut, dans un cas permettre l'obtention de bornes sur l'erreur de la solution calculée à partir de l'erreur sur les facteurs de forme, et dans l'autre cas permettre d'agir de manière *rétroactive* sous forme de contraintes sur la précision des approximations des facteurs de forme. De telles contraintes peuvent alors être utilisées par un algorithme de calcul contrôlé des facteurs de forme, *i.e* une méthode permettant d'effectuer un calcul de facteur de forme avec précision minimale.

2.3 Approche *a posteriori*

Nous présentons dans cette section la méthode d'estimation des bornes *a posteriori*, introduite par Lischinski *et al.* en 1994 [LSG94]. Le principe de cette méthode est de définir les itérations suivantes, dont la première est l'itération discrète de propagation de la lumière introduite en 1.1.4.(b).

$$\begin{aligned} B^{(n+1)} &= E + MB^{(n)} \\ \overline{B}^{(n+1)} &= E + \overline{M}\overline{B}^{(n)} \end{aligned} \quad (2.1)$$

$$\underline{B}^{(n+1)} = E + \underline{M}\underline{B}^{(n)} \quad (2.2)$$

Avec :

$$B^{(0)} = \overline{B}^{(0)} = \underline{B}^{(0)} = E$$

où les matrices \underline{M} et \overline{M} sont obtenues à partir de M en remplaçant chaque facteur de forme par des bornes respectivement inférieures et supérieure du facteur de forme :

$$\underline{F}_{ij} \leq F_{ij} \leq \overline{F}_{ij}$$

On montre alors que $\overline{B}^{(n)}$, $\underline{B}^{(n)}$ et $B^{(n)}$ vérifient :

$$\forall n \quad \underline{B}^{(n)} \leq B^{(n)} \leq \overline{B}^{(n)} \quad (2.3)$$

Nous allons maintenant rappeler deux résultats importants concernant l'obtention des bornes \underline{F}_{ij} et \overline{F}_{ij} sur le facteur de forme, ainsi que la convergence des itérations (2.1) et (2.2).

2.3.1 Obtention des bornes sur le facteur de forme

La méthode d'estimation *a posteriori* de l'erreur dont nous venons de décrire le principe requiert des bornes \underline{F}_{ij} et \overline{F}_{ij} sur le facteur de forme. Malheureusement, calculer un encadrement numériquement intéressant de la valeur exacte du facteur de forme F_{ij} entre deux surfaces A_i et A_j est un problème relativement difficile.

Lorsque la visibilité est totale, et que les surfaces sont polygonales, le calcul de telles bornes ne présente pas d'intérêt théorique puisque l'on connaît une expression analytique du facteur de forme entre deux polygones [SH93]. Il possède cependant un intérêt pratique lorsque son coût est suffisamment faible pour contre-balancer avantageusement les approximations.

Les bornes habituellement utilisées peuvent être catégorisées comme suit : on appelle *bornes exactes* (ou *conservative bounds*) les bornes mathématiquement exactes. On appelle *bornes approximatives* (ou *non conservative bounds*) celles dont le calcul n'assure pas l'exactitude, mais qui le sont dans une large mesure. Nous verrons que l'usage de telles bornes est parfois justifié.

(a) Bornes exactes

Dans les cas de visibilité totale, on obtient efficacement des bornes exactes sur le facteur de forme à partir des valeurs extrémales du facteur de forme point-polygone F_{x,A_j} . Lorsque le récepteur est polygonal, cette quantité peut être majorée à l'aide du gradient de l'irradiance aux sommets du récepteur, selon différentes configurations déterminées grâce au signe du déterminant du *Hessien* de l'irradiance en ces mêmes sommets [Hol96]. La conjecture d'unimodalité (dont nous reparlerons en 7.3.6) est invoquée pour assurer l'existence d'un unique maximum dans le cas d'une source convexe. Lorsque la source n'est pas convexe, elle est remplacée par une source équivalente convexe, aux frais d'approximations supplémentaires [Hol96]. Une méthode du même type est utilisée pour la détermination du minimum.

Lorsque la visibilité n'est que partielle, le calcul du gradient de l'irradiance devient une tâche difficile, mais réalisable [Arv94]. Ne connaissant de plus pas le nombre de ses maxima locaux, le calcul du maximum ou du minimum est alors effectué par une méthode d'optimisation, *i.e* de descente selon le gradient de l'irradiance, ce qui ne constitue plus une approche purement analytique. Le coût de ces méthodes les rend parfois inutilisables et la détermination de bornes exactes sur le facteur de forme se limite souvent dans les cas de visibilité partielle au choix des valeurs triviales 0 et F_0 , où F_0 représente le facteur de forme sans occlusion calculé analytiquement.

Les algorithmes que nous présenterons à la partie II permettent notamment d'obtenir de telles bornes.

(b) Bornes approximatives

Le principe général de détermination de bornes approximatives sur le facteur de forme dans les cas de visibilité totale ou partielle, pour une source polygonale A_j est de calculer les valeurs extrémales du facteur de forme point-polygone F_{x,A_j} sur le récepteur A_i , pour un ensemble d'échantillons x_i . On utilise pour cela des points en nombre suffisant et répartis au mieux, pour que l'encadrement suivant, par ailleurs asymptotiquement exact, soit vérifié le plus souvent possible :

$$\min_i F_{x_i,A_j} \leq F_{ij} \leq \max_i F_{x_i,A_j}$$

Dans son implémentation, Lischinski propose d'utiliser cinq points d'échantillonnage par récepteur rectangulaire : un à chaque sommet, et un au centre.

L'intérêt principal des bornes approximatives est leur faible coût, et leur précision moyenne très respectable. Leur inconvénient est qu'elles n'assurent pas d'encadrement mathématique de l'erreur d'approximation.

2.3.2 Convergence des itérations

Il est facile de voir à partir des relations (2.1) et (2.2) que chaque composante de $\underline{B}^{(n)}$ et $\overline{B}^{(n)}$ croit. Sachant que $(B^{(n)})_n$ converge, et grâce à (2.3), on sait que $(\underline{B}^{(n)})_n$ converge.

En revanche, on ne sait rien de la convergence de $(\overline{B}^{(n)})_n$. En particulier, l'itération peut diverger dès que l'on a :

$$\rho_i \sum_j \overline{F}_{ij} \geq 1$$

Pour assurer la convergence des bornes supérieures, Lischinski utilise la méthode suivante :

- trier (et renuméroter) tous les liens contribuant à la radiosité d'une feuille donnée de la hiérarchie par ordre décroissant de la radiosité sur l'émetteur.
- calculer la borne supérieure sur la radiosité de la feuille concernée grâce à l'équation suivante :

$$\overline{B}_i^{(n+1)} = E_i + \rho_i \left[\sum_{j \leq l} \overline{F}_{ij} \overline{B}_j^{(n)} + \left(1 - \sum_{j \leq l} \overline{F}_{ij} \right) \overline{B}_j^{(n)} \right]$$

L'indice l est défini comme le plus grand indice vérifiant la condition suivante après re-numérotation des indices :

$$\sum_{j \leq l} \overline{F}_{ij} < 1$$

Cette méthode assure la convergence de l'itération puisqu'elle utilise une matrice dont le rayon spectral est par construction strictement inférieur à l'unité. On montre alors que cette méthode de propagation conserve l'encadrement (2.3).

Cette méthode peut également être utilisée pour assurer la convergence de l'itération sur la radiosité elle-même, lorsque l'on n'a pas la relation suivante (Voir la section 1.2.4) :

$$\rho_i \sum_j F_{ij} < 1$$

2.3.3 Utilisation des bornes *a posteriori*

Les bornes *a posteriori* fournissent à tout instant un encadrement de la radiosité exacte (*i.e* calculée avec des facteurs de forme exacts). Elles permettent donc de connaître une borne sur l'erreur d'approximation de la solution obtenue dans l'état de convergence courant. Cet encadrement n'est cependant pas mathématiquement exact dans le cas de *bornes approximatives*, bien qu'il le soit expérimentalement la plupart du temps [LSG94].

Ces bornes peuvent également servir à la conduite du raffinement, plus précisément à choisir, à partir du moment où un lien doit être raffiné (voir la section 1.2.1.(a)), entre le raffinement de l'émetteur ou du récepteur. Pour raffiner un lien entre un émetteur A_j et un récepteur A_i , on compare l'erreur due à l'approximation du facteur de forme F_{ij} à celle due à l'approximation de la radiosité sur l'émetteur par une valeur uniforme B_j . L'erreur sur le transfert due à l'approximation de F_{ij} est mesurée par :

$$\Delta_f B_{ij} = \rho_i \overline{B}_j (\overline{F}_{ij} - \underline{F}_{ij})$$

L'erreur due à l'approximation de B_j par une valeur uniforme est, elle, mesurée par :

$$\Delta_u B_{ij} = \rho_i \bar{F}_{ij} (\bar{B}_j - \underline{B}_j)$$

Si $\Delta_f B_{ij} > \Delta_u B_{ij}$, le récepteur est raffiné, sinon on raffine l'émetteur.

Les valeurs $\bar{B}_j^{(n)}$ et $\underline{B}_j^{(n)}$ représentent à tout instant des bornes sur les valeurs extrémales de la radiosité sur A_j , mais ces bornes sont souvent trop larges car calculées après plusieurs itérations en accumulant des encadrements sur des transferts qui n'influent pas significativement sur la variation de radiosité de A_j . On utilise plutôt pour \bar{B}_j et \underline{B}_j les bornes (qui sont donc des *bornes approximatives*) obtenues en remplaçant $\bar{B}^{(n-1)}$ et $\underline{B}^{(n-1)}$ par $B^{(n-1)}$ dans le membre de droite des équations (2.1) et (2.2).

Nous utiliserons cet oracle de raffinement au chapitre 8 dans le cadre d'une méthode de radiosité hiérarchique que nous avons développée.

2.4 Approche a priori

Nous montrons dans cette section comment obtenir des bornes *a priori* sur l'erreur de la solution de radiosité calculée, en fonction des bornes sur l'erreur d'approximation des facteurs de forme. Nous discuterons ensuite de l'usage de ces bornes. L'établissement de bornes *a priori* que nous allons expliquer nécessite quelques définitions de calcul matriciel.

2.4.1 Outils mathématiques

On note φ une norme sur \mathbb{R}^n . On définit alors la norme matricielle induite par φ par

$$S_\varphi(A) = \sup_{x \neq 0} \frac{\varphi(Ax)}{\varphi(x)} \quad (2.4)$$

Il est simple de montrer à partir de (2.4) que S_φ possède les propriétés importantes suivantes :

$$S_\varphi(I) = 1 \quad (2.5)$$

$$\varphi(Ax) \leq S_\varphi(A)\varphi(x) \quad (2.6)$$

et

$$S_\varphi(AB) \leq S_\varphi(A)S_\varphi(B) \quad (2.7)$$

Si on note $\rho(A)$ le rayon spectral d'une matrice A , *i.e* le plus grand module de ses valeurs propres, on a pour toute norme φ :

$$S_\varphi(A) \geq \rho(A) \quad (2.8)$$

Nous utiliserons également le théorème suivant :

Théorème 1 (Atkinson[Atk89]) *La matrice $I - A$ est inversible à la condition suffisante² que*

$$\rho(A) < 1$$

On a alors

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k \quad \text{et pour tout } \varphi : \quad S_\varphi((I - A)^{-1}) \leq \frac{1}{1 - S_\varphi(A)}$$

²La réciproque est fautive. En revanche si la série $\sum_{k=0}^{\infty} A^k$ converge, on a bien $\rho(A) < 1$.

Pour différents choix de φ , on obtient diverses normes matricielles. On a par exemple :

$$S_{\varphi_\infty}(A) = S_\infty(A) = \max_i \sum_j |a_{ij}| \quad \text{avec} \quad \varphi_\infty(v) = \max_i |v_i| \quad (2.9)$$

On appelle enfin *conditionnement matriciel* de la matrice A pour la norme φ la quantité $\kappa_\varphi(A)$ définie par :

$$\kappa_\varphi(A) = S_\varphi(A) S_\varphi(A^{-1}) \quad \text{si } A \text{ est inversible, } +\infty \text{ sinon}$$

D'après (2.7) et (2.5), on a toujours :

$$\kappa_\varphi(A) \geq 1$$

Le conditionnement matriciel traduit de plus la distance relative d'une matrice à la matrice singulière la plus proche au sens de la norme S_φ . On a en effet [Atk89] :

$$\frac{1}{\kappa_\varphi(A)} = \inf_{\det B=0} \frac{S_\varphi(A-B)}{S_\varphi(A)}$$

Une matrice de fort conditionnement sera donc d'autant plus sensible à l'instabilité numérique lors de son inversion par des méthodes de résolution de systèmes linéaires.

Nous allons utiliser ces outils pour dégager quelques propriétés de stabilité de l'équation de radiosit .

2.4.2 Conditionnement de la matrice de radiosit 

Lors de l'application de la m thode de radiosit  classique, le syst me lin aire permettant de calculer la radiosit  sur toutes les surfaces  l mentaires s' crit

$$(I - M)B = E \quad \text{avec} \quad M = \|\rho_i F_{ij}\|_{n,n}$$

Nous avons d'autre part :

$$S_\infty(M) = \max_i \rho_i \sum_j F_{ij} \leq \underbrace{\max_i \rho_i}_{\rho_{max}} \sum_j F_{ij} \leq 1$$

Nous en d duisons, gr ce   (2.8) et au th or me 1, que :

$$\begin{aligned} S_\infty((I - M)^{-1}) &\leq \frac{1}{1 - S_\varphi(M)} \\ &\leq \frac{1}{1 - \rho_{max}} \end{aligned} \quad (2.10)$$

D'autre part,

$$\begin{aligned} S_\infty(I - M) &\leq 1 + S_\infty(M) \\ &\leq 1 + \rho_{max} \end{aligned}$$

et donc :

$$K_{\varphi_\infty}(I - M) \leq \frac{1 + \rho_{max}}{1 - \rho_{max}}$$

Ceci prouve que la matrice de radiosit  est tr s bien conditionn e, pour des sc nes   r flectance raisonnable.

2.4.3 Influence de l'erreur sur la solution

Nous allons maintenant établir une borne sur l'impact de l'approximation des facteurs de forme sur la solution du système, toujours dans le cas de la radiosité classique. Ce résultat sera ensuite étendu au cas de la radiosité hiérarchique. Nous utilisons pour cela le théorème suivant, caractérisant la solution d'un système linéaire perturbé :

Théorème 2 (Atkinson [Atk89]) Soit φ une norme sur \mathbb{R}^n et soit à résoudre le système $Ax = b$. On perturbe la matrice A en $A + \Delta A$ et le vecteur b en $b + \Delta b$, sous la condition suivante :

$$S_{\varphi}(\Delta A) < \frac{1}{S_{\varphi}(A^{-1})} \quad (2.11)$$

Alors le système perturbé admet une unique solution, notée $x + \Delta x$, vérifiant :

$$(A + \Delta A)(x + \Delta x) = b + \Delta b$$

L'écart relatif de la solution du système perturbé est alors borné selon :

$$\frac{\varphi(\Delta x)}{\varphi(x)} \leq \frac{\kappa_{\varphi}(A)}{1 - \kappa_{\varphi}(A) \frac{S_{\varphi}(\Delta A)}{S_{\varphi}(A)}} \left[\frac{S_{\varphi}(\Delta A)}{S_{\varphi}(A)} + \frac{\varphi(\Delta b)}{\varphi(b)} \right]$$

Soit ρ la matrice diagonale de terme général $\delta_{ij}\rho_i$. Notons également Δ_F la matrice des erreurs de calcul sur les facteurs de forme. Le système définissant la radiosité s'écrit alors :

$$(I - \rho(F + \Delta_F))B = E$$

Nous posons également :

$$\Delta M = \rho \Delta_F$$

Nous appliquons le théorème 2, en choisissant $A = I - M$,

$\Delta A = -\rho \Delta_F$, $b = E$, $\Delta b = 0$, et $x = B$, pour la norme $\varphi = \varphi_{\infty}$. Pour que la condition (2.11) soit vérifiée, il suffit, d'après (2.10) que :

$$S_{\infty}(\rho \Delta_F) < 1 - \rho_{\max}$$

Nous obtenons alors :

$$\varphi_{\infty}(\Delta B) \leq \varphi_{\infty}(B) \frac{\kappa_{\infty}(I - M)}{1 - \kappa_{\infty}(I - M) \frac{S_{\infty}(\rho \Delta_F)}{S_{\infty}(I - M)}} \left[\frac{S_{\infty}(\rho \Delta_F)}{S_{\infty}(I - M)} \right]$$

i.e

$$\varphi_{\infty}(\Delta B) \leq \varphi_{\infty}(B) \frac{S_{\infty}((I - M)^{-1}) S_{\infty}(\rho \Delta_F)}{1 - S_{\infty}(\rho \Delta_F) \varphi_{\infty}((I - M)^{-1})}$$

Grâce à (2.10), cette expression devient :

$$\varphi_{\infty}(\Delta B) \leq \varphi_{\infty}(B) \frac{S_{\infty}(\rho \Delta_F)}{1 - \rho_{\max} - S_{\infty}(\rho \Delta_F)}$$

Or

$$\varphi_{\infty}(B) \frac{S_{\infty}(\rho \Delta_F)}{1 - \rho_{\max} - S_{\infty}(\rho \Delta_F)} \leq \varepsilon \iff S_{\infty}(\rho \Delta_F) \leq \frac{1 - \rho_{\max}}{1 + \frac{1}{\varepsilon} \varphi_{\infty}(B)}$$

On en déduit immédiatement la condition suffisante suivante :

$$\max_i \left(\rho_i \sum_j |(\Delta_F)_{ij}| \right) \leq \frac{1 - \rho_{\max}}{1 + \frac{1}{\varepsilon} \|B\|_{\infty}} \implies \varphi_{\infty}(\Delta B) \leq \varepsilon \quad (2.12)$$

La condition (2.12) exprime effectivement la limite en dessous de laquelle nous devons cantonner l'erreur d'approximation des facteurs de forme pour que la solution calculée soit à une distance inférieure à ε du système exacte.

Nous allons maintenant discuter de l'extension de cette condition à la radiosité hiérarchique.

2.4.4 Extension à la radiosité hiérarchique

Nous reprenons les notations de la section 1.2.3, et nous notons Δ_F la matrice $n_2 \times n_1$ des erreurs sur les facteurs de forme. Le système perturbé s'écrit alors :

$$(I - \rho P(F + \Delta_F)Q)B = E$$

Il est donc toujours possible d'appliquer le théorème 2, mais en choisissant cette fois-ci :

$$\Delta A = -\rho P \Delta_F Q$$

La condition (2.11) sur la perturbation devient donc :

$$S_\infty(\rho P \Delta_F Q) < 1 - \rho_{max}$$

Et la condition obtenue s'écrit :

$$S_\infty(\rho P \Delta_F Q) \leq \frac{1 - \rho_{max}}{1 + \frac{1}{\varepsilon} \|B\|_\infty} \implies \varphi_\infty(\Delta B) \leq \varepsilon \quad (2.13)$$

Le terme $S_\infty(\rho P \Delta_F Q)$ représente la valeur maximum, calculée sur l'ensemble des feuilles A_i , de la somme des valeurs absolues des erreurs sur les facteurs de forme de tous les liens arrivant sur un parent de la feuille A_i (voir figure 1.13).

Nous avons vu dans le chapitre premier (Section 1.2.1) que dans la méthode de radiosité hiérarchique, le nombre des éléments ainsi que les facteurs de forme qui les relient varient au fil des itérations, puisque l'on raffine éventuellement ces éléments. Nous nous posons donc le problème de savoir si l'analyse que nous venons de faire reste valide au cours du raffinement.

Posons

$$S_i = \rho_i \sum_j (\rho P \Delta_F Q)_{ij}$$

On a donc

$$S_\infty(\rho P \Delta_F Q) = \max_i S_i$$

Lorsqu'on subdivise le récepteur pour un des liens arrivant sur un parent de la feuille A_i , on ne change pas le nombre de termes dans S_i puisque seuls les liens arrivant sur un parent de A_i en font partie. En revanche, si l'on suppose que l'erreur de calcul diminue pour ce lien, alors S_i diminue.

Lorsqu'on subdivise la source d'un lien arrivant sur un parent de A_i , tous les liens fils sont comptés dans la somme S_i . Lorsque l'erreur est proportionnelle à la surface de l'émetteur, comme dans le cas d'un calcul par échantillonnage, S_i ne diminue pas. Cela signifie que subdiviser les éléments source n'est d'aucun ressort dans l'augmentation de la précision selon notre étude *a priori*. Cela traduit exactement la caractéristique principale de l'analyse d'erreur *a priori* : ne prenant pas en compte la valeur de la radiosité dans la solution, elle ne peut traduire l'erreur de corrélation entre la radiosité sur l'émetteur et le noyau du facteur de forme ou la visibilité.

Finalement, sachant que le résultat du calcul itératif est indépendant du point de départ, les bornes d'erreur sur les facteurs de forme calculés après la dernière subdivision permettent de connaître grâce à (2.13) une borne sur l'erreur totale après convergence.

On ne peut donc pas utiliser une étude *a priori* sur l'erreur pour conduire le raffinement. En revanche, on peut s'en servir pour assurer une précision donnée sur la solution en agissant sur les facteurs de forme concernés.

2.4.5 Utilisation pratique

Nous nous plaçons pour commencer dans le cadre de la radiosité classique. Notons N le nombre de surfaces élémentaires, et ΔF_{max} l'erreur maximale sur les facteurs de forme. Nous avons alors :

$$\rho_i \sum_j \Delta |F_{ij}| \leq \rho_i N |\Delta F_{max}| \leq N \rho_{max} |\Delta F_{max}| \quad (2.14)$$

D'où la condition plus forte :

$$\Delta F_{max} \leq \frac{1 - \rho_{max}}{N \rho_{max}} \frac{1}{1 + \frac{\|B\|_{\infty}}{\epsilon}} \implies \|\Delta B\|_{\infty} \leq \epsilon \quad (2.15)$$

Exemple

Nous choisissons les valeurs suivantes, représentatives d'une simulation dans une scène de moyenne complexité : $\rho_{max} = 0.6$, $N = 1000$, $\frac{\epsilon}{\|B\|_{\infty}} = 5\%$. Nous obtenons alors la condition

$$\Delta F_{max} \leq 3.10^{-5}$$

(a) Algorithme pratique

Au vu de l'exemple que nous venons de décrire, la contrainte de précision sur les facteurs de forme apparaît comme assez forte. Cela vient du fait que la majoration (2.14) suppose que l'erreur maximale ΔF_{max} peut être atteinte par tous les facteurs de forme. La déviation de la solution est donc virtuellement considérée comme résultant de l'accumulation de toutes ces erreurs potentielles.

Une utilisation plus intelligente des conditions (2.12) et (2.13) correspondrait par exemple à comptabiliser les erreurs sur les facteurs de forme contribuant à la radiosité d'une feuille donnée (une ligne de la matrice Δ_F en radiosité classique, et $P\Delta_F Q$ en radiosité hiérarchique) et imposer que leur somme vérifie la condition (2.12) (ou (2.13)). Cela permettrait une bien meilleure répartition des erreurs, sachant que pour la plupart des éléments de la ligne, l'erreur de calcul est nulle (par exemple dans les cas de visibilité totale) ou très en dessous du seuil imposé par (2.15). Nous n'avons pas encore exploré cette voie, qui paraît relativement prometteuse.

2.5 Discussion et conclusion

Nous avons vu deux méthodes d'estimation de l'erreur en radiosité hiérarchique : l'une *a posteriori* et l'autre *a priori*, toutes deux basées sur un encadrement de l'erreur de calcul du facteur de forme. L'avantage des bornes *a posteriori* est qu'elles sont en général assez fiables puisqu'elles prennent spécifiquement en compte la solution calculée. En revanche elles ne sont pas compatibles avec le contrôle de l'erreur puisqu'elles ne permettent pas d'en connaître la cause. A l'inverse, les bornes *a priori* offrent la possibilité d'un contrôle de l'erreur sur la solution, mais sont généralement plus pessimistes, puisque dans une large mesure indépendantes des conditions initiales.

La méthode d'estimation de l'erreur *a posteriori* introduite par Lischinski *et al.*, ainsi que la méthode d'estimation *a priori* que nous avons proposée, motivent le développement d'un algorithme de contrôle de l'erreur dans le calcul des transferts d'énergie, c'est à dire un algorithme capable d'assurer une précision donnée sur le résultat du calcul du facteur de forme. L'usage d'un tel outil dépend cependant du type d'estimation de l'erreur utilisé : dans le cadre d'une estimation *a priori* de l'erreur, l'utilisateur impose une précision donnée sur la solution, laquelle se traduit par un critère de précision sur le calcul du facteur de

forme, que doit respecter l'algorithme de calcul contrôlé. S'il s'agit d'une estimation *a posteriori*, l'algorithme de calcul contrôlé du facteur de forme retourne la meilleure valeur obtenue accompagnée de bornes d'erreur qui sont ensuite utilisées pour borner l'erreur sur la solution.

Nous expliquerons aux chapitres 3, 4 et 5 un système de calcul contrôlé du facteur de forme permettant la production de bornes conservatives en radiosité hiérarchique.



Au chapitre 2, nous avons étudié l'impact des approximations des facteurs de forme sur la solution de l'équation discrète de radiosité. Nous avons vu d'une part qu'assortir le calcul des facteurs de forme d'une borne sur l'erreur d'approximation permet d'effectuer une analyse a posteriori de l'erreur sur la solution du système linéaire. Nous avons montré d'autre part, qu'une analyse a priori de l'erreur globale sur la solution peut permettre d'obtenir une solution réalisant une précision imposée à l'avance, si l'on dispose d'un algorithme de calcul des facteurs de forme capable de fournir une approximation de chaque facteur de forme avec une précision fixée. Dans la deuxième partie de ce document, nous allons développer de tels algorithmes, basés sur une formulation hiérarchique de la visibilité. Le chapitre 3 expose le principe général de ces algorithmes. Nous en proposerons ensuite des implémentations dans le cadre de la radiosité en dimension deux et trois aux chapitres 4 et 5.

Deuxième partie

**Calcul contrôlé des facteurs de
forme**

Chapitre 3

Contrôle de l'erreur d'approximation de la visibilité dans le calcul des facteurs de forme

UN ALGORITHME de calcul contrôlé des facteurs de forme est une méthode permettant de calculer un facteur de forme assorti de bornes sur l'erreur d'approximation, ou encore d'effectuer ce calcul le plus rapidement possible, tout en assurant que le résultat ne diffère pas plus qu'une borne imposée, de la valeur exacte du facteur de forme.

Nous avons pour cela construit une méthode d'approximation des facteurs de forme basée sur le pré-calcul d'un certain nombre d'informations de visibilité dans la hiérarchie de tous les obstacles possibles [SS96b, SS96a]. Cette information est ensuite utilisée dans chaque calcul de facteur de forme pour accéder à la fois à une représentation approximative du facteur de forme, et à une borne sur l'erreur d'approximation, grâce à un algorithme hiérarchique.

Nous resterons, dans ce chapitre, volontairement très généraux quant aux détails liés à l'implémentation pratique des algorithmes développés, sachant que nous présenterons au chapitre 4 l'implémentation de ces algorithmes dans le cadre de la radiosité en dimension deux, et discuterons au chapitre 5 de leur application au cas tri-dimensionnel.

3.1 Définitions

Nous introduisons dans cette section les notions de *mesure d'un ensemble de droites de l'espace*, *transmittance* et *transmittance directionnelle*, utilisées tout au long de ce chapitre.

Définition 7 *Il existe en deux (resp. en trois) dimensions, une unique mesure définie à un facteur multiplicatif près, sur l'espace des droites, qui soit invariante par le groupe des déplacements du plan (resp. de l'espace) [San76].*

Nous utiliserons la notation suivante, indépendante de la paramétrisation et de la dimension de l'espace, pour exprimer la mesure d'un ensemble S de droites :

$$\int_S dL$$

De même, la mesure des droites rencontrant un ensemble de points S sera notée :

$$\int_{L \cap S \neq \emptyset} dL$$

Nous verrons par ailleurs au chapitre 4 l'expression pratique d'une telle mesure, en dimension deux.

Définition 8 On appelle transmittance d'un objet ou d'un groupe d'objets le long d'une droite L , le coefficient $\tau(L)$ par lequel est multipliée l'intensité lumineuse d'un rayon représenté par L et traversant l'objet :

$$I_{out}(L) = \tau(L)I_{in}(L)$$

Pour des objets opaques, la transmittance vaut donc 0 ou 1 selon que le rayon porté par L rencontre l'objet ou pas. Pour des milieux semi-transparentes, il est toujours possible de définir la transmittance de cette façon pourvu que la lumière traverse le milieu sans changement de direction ni diffusion. Elle est alors représentée par un réel de l'intervalle $[0, 1]$. Enfin, étant définie sur l'ensemble des droites de l'espace, la transmittance est une fonction de quatre variables en dimension trois, et de deux variables en dimension deux.

Définition 9 La transmittance directionnelle d'un objet ou d'un groupe d'objets dans la direction d est la moyenne $\bar{\tau}(d)$ de sa transmittance, calculée sur l'ensemble des droites parallèles d qui rencontrent sa boîte englobante.

Dans cette définition, nous appelons *boîte englobante* d'un objet ou d'un groupe d'objets, le plus petit parallélépipède aux cotés parallèles aux axes du repère, le contenant. Dans la pratique, la transmittance directionnelle peut donc être mesurée en projetant orthogonalement l'objet considéré, ainsi que sa boîte englobante, sur un hyperplan normal à d . La quantité $\bar{\tau}(d)$ vaut alors le complément à l'unité, du rapport des aires des projections de l'objet et de sa boîte englobante.

Comme son nom l'indique, la transmittance directionnelle est une *fonction directionnelle*, i.e définie sur l'ensemble des directions. Elle prend donc la forme d'une fonction de deux variables en dimension trois et d'une seule variable en dimension deux.

3.1.1 Expression du facteur de forme en terme de mesures de droites

Comme nous l'avons expliqué au chapitre 1, le facteur de forme entre deux surfaces S_i et S_j représente la fraction d'énergie lumineuse qui, quittant S_i , atteint S_j . Dans le vide et, en première approximation, dans l'air la lumière se propage en ligne droite. D'autre part sous les hypothèses du modèle Lambertien, l'intensité de l'énergie lumineuse quittant ou rencontrant une surface en un point donné ne dépend pas de la direction incidente du rayon lumineux considéré à la surface. La proportion d'énergie lumineuse qui, quittant une surface S_i atteint une surface S_j vaut donc le rapport entre la mesure des droites qui coupent S_i et S_j , et la mesure de celles qui coupent S_i :

$$F_{ij} = \frac{\int_{(L \cap S_i \neq \emptyset) \wedge (L \cap S_j \neq \emptyset)} dL}{\int_{L \cap S_i \neq \emptyset} dL}$$

Nous verrons au chapitre 4 comment s'écrit cette expression dans le cas de la radiosité en dimension deux.

En présence d'obstacles, il faut soustraire de l'ensemble des droites considérées, celles qui rencontrent un des obstacles. En notant C l'ensemble des obstacles situés entre S_i et S_j , on a alors :

$$F_{ij} = \frac{1}{\int_{L \cap S_i \neq \emptyset} dL} \left[\int_{(L \cap S_i \neq \emptyset) \wedge (L \cap S_j \neq \emptyset) \wedge (L \cap C = \emptyset)} dL \right]$$

Cette formulation en termes de mesures de droites a été utilisée par Sbert [Sbe93] et Pellegrini [Pel95] pour le calcul de facteurs de forme par des méthodes d'échantillonnage aléatoire. Nous nous en servirons à la section 3.4 pour exprimer le facteur de forme en présence d'obstacles, en fonction de facteurs de forme considérant des sous-parties de ces obstacles.

3.2 Contrôle de l'erreur et information multi-échelles de visibilité

Nous nous posons le problème d'obtenir une approximation contrôlée du facteur de forme entre deux surfaces S_1 et S_2 , en présence d'obstacles. Nous supposons de plus, que ces obstacles sont regroupées en une hiérarchie de *clusters* dont la racine sera notée C_0 . Nous cherchons donc à construire un algorithme fournissant de manière plus ou moins rapide une approximation \tilde{F}_{C_0} du facteur de forme (noté F_{C_0}), pour laquelle l'erreur d'approximation est inférieure à une borne ε fixée par l'utilisateur, *i.e* :

$$|\tilde{F}_{C_0} - F_{C_0}| \leq \varepsilon$$

Nous proposons pour cela d'équiper la hiérarchie des obstacles, d'informations donnant, pour tout obstacle C , un accès direct à une approximation \overline{F}_C , du facteur de forme F_C entre les deux surfaces à partir d'une formulation simplifiée de la visibilité à travers C , et donnant également accès à un majorant de l'erreur d'approximation.

Une borne sur l'erreur due à l'approximation de F_C par \overline{F}_C doit donc dépendre en partie des propriétés géométriques propres à la hiérarchie des objets occultants (répartition relatives de ces objets, uniformité et densité) mais également de la configuration courante du calcul, qui ne concerne plus uniquement les obstacles, mais aussi les deux surfaces : leur taille et orientation relative et leur radiosité. Nous noterons donc une telle borne d'erreur sous une forme visant à faire apparaître cette double dépendance. Pour des surfaces S_1 et S_2 et un obstacle C , nous désignerons cette borne d'erreur d'approximation de la visibilité par :

$$\text{BVE}(S_1, S_2, C)$$

Pour une approximation \overline{F}_C du facteur de forme F_C , la borne d'erreur BVE doit donc vérifier :

$$|F_C - \overline{F}_C| \leq \text{BVE}(S_1, S_2, C)$$

Disposant pour tout obstacle potentiel C d'une borne sur l'erreur due à l'approximation de F_C par \overline{F}_C , nous proposons de calculer une approximation contrôlée du facteur de forme grâce à l'algorithme récursif suivant :

- Si l'approximation \overline{F}_C est suffisante, *i.e* si le critère d'erreur vérifie $\text{BVE}(S_1, S_2, C) \leq \varepsilon$, alors \overline{F}_C est choisi pour approcher F_C , et nous posons donc $\tilde{F}_C = \overline{F}_C$;
- sinon, nous calculons par un appel récursif, des approximations contrôlées $\tilde{F}_{C_1}, \tilde{F}_{C_2}, \dots, \tilde{F}_{C_n}$ des facteurs de forme F_{C_1}, \dots, F_{C_n} entre S_1 et S_2 , en ne considérant séparément comme obstacle qu'un seul fils C_i de C , et en utilisant des valeurs de contrôle ε_i appropriées. Ces approximations sont ensuite *combinées* pour calculer une approximation de F_C , dont l'erreur est bornée par ε .

Cet algorithme est donc basé sur le pré-calcul et le stockage pour chaque obstacle, d'une certaine quantité d'informations, qui peuvent ensuite être ré-utilisées pour approcher le facteur de forme, et mesurer l'erreur d'approximation, dans n'importe quelle configuration de calcul de facteur de forme.

Pour mettre en place un tel algorithme, nous devons donc résoudre les problèmes suivants :

- définir une méthode efficace d'approximation de la visibilité dans le calcul du facteur de forme ainsi qu'une d'une borne sur l'erreur d'approximation, *i.e* savoir calculer \overline{F}_C pour tout C ainsi qu'une borne supérieure de la quantité $|\overline{F}_C - F_C|$.
- définir, calculer et stocker l'information nécessaire à l'obtention de ces quantités à tous les niveaux de la hiérarchie des obstacles.
- savoir *combiner* des approximations contrôlées $\tilde{F}_{C_1}, \tilde{F}_{C_2}, \dots, \tilde{F}_{C_n}$ des facteurs de forme F_{C_1}, \dots, F_{C_n} (obtenues en ne considérant séparément qu'un seul sous-obstacle de l'obstacle C) en une estimation du facteur de forme considérant l'obstacle C en entier.

- déterminer les valeurs $\varepsilon_1, \dots, \varepsilon_n$ contraignant le calcul de $\tilde{F}_{C_1}, \dots, \tilde{F}_{C_n}$ de façon à ce que la combinaison \tilde{F}_C de ces différents facteurs de forme vérifie le critère de précision $|\tilde{F}_C - F_C| \leq \varepsilon$.

Nous définissons l'*information multi-échelles de visibilité* comme étant l'ensemble des quantités dont le pré-calcul et le stockage dans la hiérarchie des obstacles fournissent l'approximation \bar{F}_C ainsi que la borne BVE sur l'erreur associée.

Nous traiterons à la section 3.3 du choix d'une approximation \bar{F}_C . A la section 3.4, nous expliquerons comment *combiner* les facteurs de forme utilisant chaque sous-obstacle, avant de détailler, en 3.5, l'algorithme proprement dit.

3.3 Approximation du facteur de forme

Nous voulons pouvoir obtenir une approximation \bar{F}_C du facteur de forme F_C entre deux surfaces S_1 et S_2 en considérant un obstacle C , à partir d'informations pré-calculées. Le principe de notre méthode est de concentrer l'effort dans la partie concernant le pré-calcul de l'information, afin de permettre ensuite dans n'importe quelle configuration, de calculer en temps constant l'approximation \bar{F}_C et la borne BVE.

Reprenons par exemple l'expression (1.9) du facteur de forme, que l'on applique entre deux surfaces S_1 et S_2 (s_1 désigne l'aire de S_1) :

$$F_{12} = \frac{1}{s_1} \int_{S_1} \int_{S_2} k(x,y)v(x,y)dx dy$$

En approchant le terme de visibilité $v(x,y)$ par une valeur uniforme \bar{v} sur le domaine d'intégration, nous avons :

$$F_{12} \# \underbrace{\bar{v} \frac{1}{s_1} \int_{S_1} \int_{S_2} k(x,y)dx dy}_{F_0}$$

Dans cette expression, la quantité F_0 représente donc le facteur de forme *sans occlusion* entre S_1 et S_2 . Nous obtenons alors une borne sur l'approximation ainsi réalisée en trouvant un majorant de la différence entre les deux expressions. Un choix possible pour \bar{v} est la transmittance directionnelle $\bar{\tau}_C$ de l'obstacle C dans une direction d_{12} , représentative du domaine d'intégration $S_1 \times S_2$. Dans ce cadre, l'information multi-échelles de visibilité se composera pour l'obstacle C , de la fonction $\bar{\tau}_C$ et de diverses autres quantités permettant de borner la différence $|F_C - \bar{\tau}(d_{12})F_0|$. Elle sera donc pré-calculée et stockée une fois pour toutes.

Nous proposerons aux chapitres 4 et 5 plusieurs méthodes permettant le calcul de telles approximations ainsi que des bornes d'erreur associées, dans le cadre de la radiosité en dimensions deux et trois. Nous pouvons cependant d'ores et déjà dégager quelques propriétés de l'information de visibilité qui seront requises pour le bon fonctionnement de l'algorithme de calcul contrôlé des facteurs de forme :

Pour permettre le calcul hiérarchique contrôlé du facteur de forme, la borne d'erreur BVE doit décroître lorsque l'on subdivise l'obstacle, et cela en tenant compte de l'erreur éventuellement produite par la combinaison des sous facteurs de forme. Nous devons donc avoir :

$$\text{BVE}(S_1, S_2, C) \geq \text{Combine}(\text{BVE}(S_1, S_2, C_1), \dots, \text{BVE}(S_1, S_2, C_n))$$

l'expression **Combine** traduisant le calcul de l'erreur d'approximation après combinaison des facteurs de forme approchés aux niveaux inférieurs.

Selon le type d'approximation réalisée, l'information multi-échelles de visibilité prend la forme de constantes ou de fonctions directionnelles. Le choix de la représentation sera le résultat d'un compromis entre la qualité des approximations et des bornes produites en fonction des caractéristiques des obstacles et la taille mémoire de ces représentations : Représenter l'information multi-échelles de visibilité par des constantes revient à supposer une totale isotropie des obstacles, ce qui n'est pas toujours le cas, comme le montre la figure 3.1. En revanche, calculer et stocker la transmittance exacte des obstacles permettrait des calculs arbitrairement précis mais le coût de stockage en serait prohibitif.

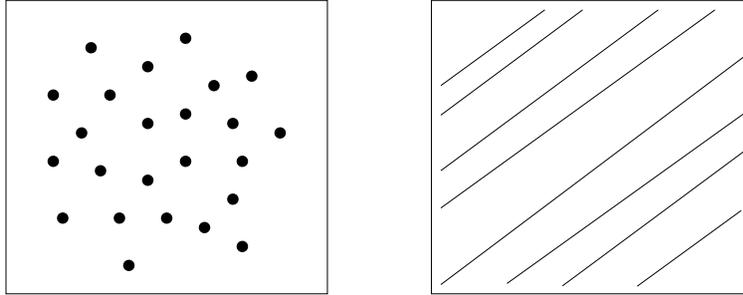


FIG. 3.1: Exemples d'obstacle isotropes et anisotropes. Alors qu'une unique valeur suffit à représenter la visibilité moyenne de manière raisonnablement précise (sur un domaine d'intégration grand devant la taille des objets) pour un cluster isotrope (à gauche), une fonction dépendant au moins de la direction de vue est nécessaire pour les obstacles fortement anisotropes (à droite).

3.4 Combinaison des facteurs de forme

Lorsque l'approximation \bar{F}_C du facteur de forme, utilisant l'information de visibilité d'un obstacle C , s'avère insuffisante au vu de la précision requise ε , l'algorithme que nous avons présenté *raffine* l'obstacle en calculant séparément des approximations $\tilde{F}_{C_1}, \dots, \tilde{F}_{C_n}$ des facteurs de forme F_{C_1}, \dots, F_{C_n} entre S_1 et S_2 en considérant chaque fils C_i de C comme unique obstacle. Nous nous posons donc le problème de savoir calculer une approximation de F_C et un encadrement de l'erreur d'approximation à partir de ces seules valeurs.

3.4.1 Formule de combinaison

Comme nous l'avons vu à la section 3.1.1, le facteur de forme F_C entre deux surfaces S_1 et S_2 , et en présence d'un obstacle C , peut être exprimé comme :

$$F_C = \frac{1}{\int_{L \cap S_1 \neq \emptyset} dL} \left[\int_{(L \cap S_1 \neq \emptyset) \wedge (L \cap S_2 \neq \emptyset) \wedge (L \cap C) = \emptyset} dL \right]$$

Dans cette expression $L \cap C$ désigne l'intersection du rayon L avec l'ensemble des objets contenus dans C (et non la boîte englobante de ces objets, par exemple). Notons avec la même convention $\mu(S_1, S_2, C_1, \dots, C_k)$ la mesure des droites rencontrant S_1 , S_2 ainsi que C_1, \dots, C_k . Nous avons alors :

$$F_C = \frac{1}{\mu(S_1)} [\mu(S_1, S_2) - \mu(S_1, S_2, C)] \quad (3.1)$$

Lorsque qu'aucun obstacle n'existe entre S_1 et S_2 , nous retrouvons l'expression du facteur de forme *sans occlusion* :

$$F_0 = \frac{\mu(S_1, S_2)}{\mu(S_1)}$$

Supposons maintenant que les sous-obstacles C_i de C sont disjoints deux à deux. Nous pouvons alors utiliser la propriété suivante, donnant l'expression de la mesure d'une réunion de n ensembles E_1, \dots, E_n :

$$\begin{aligned} \mu(\cup_{i=1}^n E_i) &= \sum_{i=1}^n \mu(E_i) - \sum_{1 \leq i_1 < i_2 \leq n} \mu(E_{i_1} \cap E_{i_2}) + \sum_{1 \leq i_1 < i_2 < i_3 \leq n} \mu(E_{i_1} \cap E_{i_2} \cap E_{i_3}) - \dots \\ &\quad + (-1)^{n+1} \mu\left(\bigcap_{i=1}^n E_i\right) \end{aligned}$$

A partir de (3.1), nous pouvons donc écrire :

$$\begin{aligned} \mu(S_1)F_C &= \mu(S_1, S_2) - \sum_{i_1=1}^n \mu(S_1, S_2, C_{i_1}) + \sum_{1 \leq i_1 < i_2 \leq n} \mu(S_1, S_2, C_{i_1}, C_{i_2}) + \dots \\ &= (1-n)\mu(S_1, S_2) + \sum_{i_1=1}^n \underbrace{\mu(S_1, S_2) - \mu(S_1, S_2, C_{i_1})}_{\mu(S_1)F_{C_{i_1}}} + \dots \end{aligned}$$

et donc finalement :

$$F_C = \sum_{i=1}^n F_{C_i} - (n-1)F_0 + \chi$$

avec

$$\chi = \frac{1}{\mu(S_1)} \left(\sum_{1 \leq i_1 < i_2 \leq n} \mu(S_1, S_2, C_{i_1}, C_{i_2}) - \sum_{1 \leq i_1 < i_2 < i_3 \leq n} \mu(S_1, S_2, C_{i_1}, C_{i_2}, C_{i_3}) + \dots \right)$$

Dans le cas de deux sous-clusters, la formule ci-dessus se simplifie considérablement :

$$F = F_{C_1} + F_{C_2} - F_0 + \chi \quad \text{avec} \quad \chi = \frac{\mu(S_1, S_2, C_1, C_2)}{\mu(S_1)} \quad (3.2)$$

Le terme χ est un terme de *corrélation* entre les sous-obstacles C_i . Il représente une interaction complexe entre les fonctions de transmittance des différents sous-obstacles. Nous traiterons à la section 4.3 du calcul d'une borne sur cette quantité en dimension deux.

La figure 3.2 montre combien la corrélation des sous-obstacles est importante dans le calcul récursif des facteurs de forme. Dans l'exemple proposé, les deux sous-obstacles, formés de grilles laissant passer environ 50 % de la lumière, les facteurs de forme F_1 et F_2 sont donc tous deux proches de $\frac{1}{2}F_0$. En revanche, selon la position relative des sous-obstacles C_1 et C_2 , F_C varie de 0 à $\frac{1}{2}F_0$.

Cet exemple montre donc qu'il est impossible de se passer de la corrélation des sous-obstacles dans le calcul récursif des facteurs de forme. Nous verrons à la section 3.6 que la corrélation joue le même rôle dans le calcul récursif de l'information multi-échelles de visibilité.

3.4.2 Application au calcul récursif contrôlé

Plaçons-nous pour simplifier dans le cas de deux sous-obstacles. La formule (3.2) nous permet de calculer le facteur de forme F_C en fonction de F_{C_1} et F_{C_2} . Nous nous intéressons maintenant au calcul d'une borne sur l'erreur d'approximation associée à ce calcul.

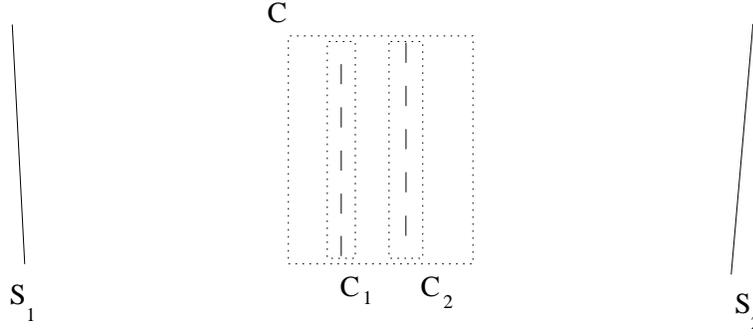


FIG. 3.2: Exemple de situation dans laquelle la corrélation des sous-obstacles peut jouer un rôle important (voire texte).

Supposons connaître un encadrement de la corrélation χ entre les sous clusters C_1 et C_2 dans le domaine d'intégration concerné :

$$\chi_{min} \leq \chi \leq \chi_{max}$$

Soit ε_1 et ε_2 des bornes d'erreur obtenues sur le calcul de F_{C_1} et F_{C_2} . Nous choisissons alors l'approximation suivante pour F_C :

$$F_C \approx F_{C_1} + F_{C_2} - F_0 + \frac{1}{2} (\chi_{min} + \chi_{max})$$

Une borne d'erreur sur l'approximation réalisée est alors :

$$E = \varepsilon_1 + \varepsilon_2 + \varepsilon_{Correl} \quad \text{avec} \quad \varepsilon_{Correl} = \frac{1}{2} (\chi_{max} - \chi_{min}) \quad (3.3)$$

Nous discuterons de l'estimation de la corrélation entre sous-obstacles au chapitre 4. Nous retiendrons pour l'instant que cela peut nécessiter le pré-calcul et stockage d'informations supplémentaires.

3.5 Algorithme de calcul récursif du facteur de forme

Nous décrivons dans cette section l'algorithme de calcul récursif du facteur de forme entre deux surfaces S_1 et S_2 occultées par une hiérarchie de clusters de racine C , selon le principe général établi à la section 3.2 et en utilisant les méthodes de combinaison des facteurs de forme et des erreurs, décrites en 3.4. Pour simplifier, nous supposons, sans perte de généralité, que la hiérarchie des obstacles est binaire.

Lorsque la borne $BVE(S_1, S_2, C)$ indique que l'erreur d'approximation du facteur de forme par \bar{F}_C est inférieure au seuil ε_{max} fixé par l'utilisateur, l'algorithme se satisfait de cette valeur. Dans le cas contraire, l'algorithme tente de subdiviser l'obstacle en utilisant la formule de combinaison décrite à la section précédente : deux facteurs de forme sont calculés en considérant séparément chaque sous-cluster de C comme unique obstacle et en imposant des seuils d'erreur propres ε_1 et ε_2 tels que l'erreur totale associée à la valeur combinée reste bornée par ε . Nous discutons du choix de ε_1 et ε_2 en (a).

La formule de combinaison des erreurs (3.3) indique que l'erreur cumulée d'une combinaison d'approximations de facteurs de forme n'est pas la somme des erreurs de chaque facteur de forme. Il faut en effet rajouter l'erreur d'approximation de la corrélation de visibilité entre les sous-obstacles. Lorsque cette dernière composante est trop importante, on ne peut alors pas subdiviser l'obstacle. Lorsque la combinaison n'est pas possible, la seule façon de respecter le seuil d'erreur est d'effectuer un calcul précis, et donc coûteux, du facteur de forme. Nous en discuterons en (b).

La figure 3.3 donne le pseudo-code reproduisant notre méthode récursive de calcul contrôlé du facteur de forme. La procédure **RFormFactor** effectue l'approximation contrôlée du facteur de forme. La fonction **HFormFactor** est la fonction appelée lorsque la subdivision de l'obstacle ne permet pas d'augmenter la précision à cause de la corrélation des sous-obstacles.

```

RFormFactor(element  $S_1$ , element  $S_2$ , Cluster  $C$ , float  $\epsilon_{max}$ , outfloat  $F$ )
  If  $BVE(S_1, S_2, C) \leq \epsilon$ 
     $F = \bar{F}_C$ 
  Else
    If  $\epsilon_{Correl} \leq \epsilon$ 
      RFormFactor( $S_1, S_2, C_1, \epsilon_1^{max}, F_{C_1}$ )
      RFormFactor( $S_1, S_2, C_2, \epsilon_1^{max}, F_{C_2}$ )
       $F = F_{C_1} + F_{C_2} - F_0 + \frac{1}{2}(\chi_{min} + \chi_{max})$ 
    Else
       $F = \mathbf{HFormFactor}(S_1, S_2, C)$ 

```

FIG. 3.3: Pseudo-code pour le calcul contrôlé du facteur de forme.

(a) Répartition des seuils d'erreur

Comme l'indique la formule (3.3), les bornes d'erreur s'ajoutent lors du calcul récursif et produisent une borne associée au calcul courant après prise en compte d'une troisième borne d'erreur sur l'approximation de la corrélation. Pour assurer que l'erreur totale restera inférieure au seuil ϵ_{max} lors de la subdivision de l'obstacle, nous pouvons donc partager ϵ_{max} entre l'erreur de corrélation ϵ_{Correl} , dont on suppose connaître une borne, et les seuils utilisés dans les appels récursifs qui eux, sont choisis. Connaissant une borne ϵ_{Correl} sur l'erreur de corrélation, un choix simple est d'utiliser les plus grandes valeurs possibles ϵ_1 et ϵ_2 vérifiant :

$$\epsilon_1 + \epsilon_2 + \epsilon_{Correl} \leq \epsilon_{max}$$

Ce qui signifie :

$$\epsilon_1 = \epsilon_2 = \frac{1}{2}(\epsilon_{max} - \epsilon_{Correl}) \quad (3.4)$$

Alors que, par construction, l'algorithme présenté fournit alors une approximation du facteur de forme dont l'erreur est inférieure à ϵ_{max} , rien ne lui permet de prendre en compte la "largeur" des bornes dans le calcul. Quelle que soit la précision avec laquelle les appels récursifs calculent une approximation du facteur de forme, l'erreur associée à la combinaison sera toujours en retour de procédure égale à ϵ_{max} .

Expérimentalement, nous avons remarqué que l'erreur effectivement associée au calcul est parfois bien plus faible que l'erreur ϵ allouée par ce calcul. Celle-ci vaut par exemple au plus $BVE(S_1, S_2, C)$ quand $BVE(S_1, S_2, C) \leq \epsilon_{max}$, 0 dans le cas d'un appel au calcul exact. En revanche, en utilisant les valeurs de l'équation (3.4) on montre facilement qu'à la profondeur n des appels récursifs, on choisit obligatoirement :

$$\epsilon_1^{(n)} = \epsilon_2^{(n)} \leq \frac{1}{2^n} \epsilon_{max}^{(0)}$$

D'autre part, nous avons observé que, dans la pratique, χ décroît proportionnellement au diamètre des obstacles. On a donc très rapidement $\epsilon_{Correl} \geq \epsilon_{max}^{(n)}$, et l'algorithme explose à cause d'un trop grand nombre d'appels au calcul exact.

3.5. ALGORITHME DE CALCUL RÉCURSIF DU FACTEUR DE FORME 65

Nous proposons donc de gérer la répartition des seuils d'erreur de manière plus fine en calculant explicitement l'erreur ε associée au calcul courant :

- ε vaut $\text{BVE}(S_1, S_2, C)$ si $\text{BVE}(S_1, S_2, C) \leq \varepsilon$
- ε vaut 0 en cas de calcul exact
- ε vaut $\varepsilon_1 + \varepsilon_2 + \varepsilon_{\text{Correl}}$ en cas d'appels récursifs. Dans ce cas, ε_1 et ε_2 désignent les erreurs ainsi calculées dans les appels récursifs.

Pour respecter le seuil ε_{max} , nous utilisons une répartition de l'erreur suivante : le premier appel récursif à pour valeur de seuil un nombre $\varepsilon_1^{\text{max}}$ pouvant varier entre 0 et $\varepsilon_{\text{max}} - \varepsilon_{\text{Correl}}$. Au cours de cet appel, une valeur d'erreur effective ε_1 est calculée. On utilise alors dans le second appel récursif la totalité de la marge d'erreur restante comme seuil, c'est à dire :

$$\varepsilon_2^{\text{max}} = \varepsilon_{\text{max}} - \varepsilon_1 - \varepsilon_{\text{Correl}}$$

Au cours de ce second appel est calculée une erreur effective ε_2 . L'erreur effective calculée au niveau courant sera donc

$$\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_{\text{Correl}}$$

Le choix de $\varepsilon_1^{\text{max}}$ est assez délicat : les valeurs extrêmes 0 et $\varepsilon_{\text{max}} - \varepsilon_{\text{Correl}}$ sont clairement inadaptées. Nous n'avons pas non plus choisi la valeur médiane de l'intervalle des valeurs possibles car elle ne tient pas compte de l'ordre des appels récursifs. Nous avons déterminé empiriquement que la valeur suivante permettait une répartition en moyenne équitable de l'erreur sur les appels :

$$\varepsilon_1^{\text{max}} = \frac{3}{4}(\varepsilon_{\text{max}} - \varepsilon_{\text{Correl}})$$

Pour des raisons équivalentes, il n'est pas non plus efficace de beaucoup relaxer la contrainte sur l'erreur de corrélation car cela réduit trop le seuil d'erreur des appels récursifs. Nous avons empiriquement choisi la valeur de seuil de $\frac{\varepsilon_{\text{max}}}{2}$. Bien que ces choix améliorent très sensiblement les performances de la méthode, il s'agit toujours d'une répartition arbitraire des seuils d'erreur et qui n'est donc meilleure qu'en moyenne. L'algorithme obtenu est présenté sur la figure 3.4.

```

RFormFactor(element  $S_1$ , element  $S_2$ , Cluster  $C$ , float  $\varepsilon_{\text{max}}$ , out float  $\varepsilon$ , out float  $F$ )
  If  $\text{BVE}(S_1, S_2, C) \leq \varepsilon$ 
     $F = \overline{F}_C$ 
     $\varepsilon = \text{BVE}(S_1, S_2, C)$ 
  Else
    If  $\varepsilon_{\text{Correl}} \leq \frac{\varepsilon_{\text{max}}}{2}$ 
      float  $\varepsilon_1, F_{C_1}, F_{C_2}$ 
      float  $\varepsilon_1^{\text{max}} = \frac{3}{4}(\varepsilon_{\text{max}} - \varepsilon_{\text{Correl}})$ 
      RFormFactor( $S_1, S_2, C_1, \varepsilon_1^{\text{max}}, \varepsilon_1, F_{C_1}$ )
      float  $\varepsilon_2^{\text{max}} = \varepsilon_{\text{max}} - \varepsilon_1 - \varepsilon_{\text{Correl}}$ 
      RFormFactor( $S_1, S_2, C_2, \varepsilon_2^{\text{max}}, \varepsilon_1, F_{C_1}$ )
       $F = F_{C_1} + F_{C_2} - F_0 + \frac{1}{2}(\chi_{\text{min}} + \chi_{\text{max}})$ 
       $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_{\text{Correl}}$ 
    Else
       $F = \text{HFormFactor}(S_1, S_2, C)$ 
       $\varepsilon = 0$ 

```

FIG. 3.4: Calcul récursif du facteur de forme avec répartition des seuils d'erreur.

(b) Calcul exact du facteur de forme

Dans le cas où l'erreur de corrélation ne permet pas de fournir une approximation du facteur de forme d'erreur inférieure au seuil fixé, l'algorithme que nous avons proposé fait appel à un calcul "arbitrairement précis" du facteur de forme. Lors de l'implémentation de notre algorithme, nous avons en fait plusieurs alternatives, selon le contexte d'utilisation du calcul contrôlé du facteur de forme :

Dans le cadre d'une méthode *a posteriori* de contrôle de la précision, comme celle que nous avons présentée au chapitre 2, nous acceptons le surplus d'erreur due à la corrélation (qui ne concerne souvent qu'une partie du résultat), et l'algorithme effectue ses appels récursifs, mais n'assure plus que $\varepsilon < \varepsilon_{max}$. L'erreur calculée en même temps que le facteur de forme peut alors servir à l'encadrement du résultat.

En revanche, dans le cadre d'une méthode *a priori* de contrôle de la précision (voire également le chapitre 2, le seuil sur l'erreur est une borne absolue. On est alors obligé de considérer le temps de calcul : si l'on ne peut pas se permettre un coût trop élevé, une exception est levée et l'algorithme abandonne son calcul ; si, en revanche, le temps de calcul est illimité, l'appel à *HFormFactor* s'effectue normalement. Il faut cependant remarquer que le coût d'un "calcul exact" de facteur de forme à une certaine profondeur dans les appels récursifs est toujours inférieur à celui du calcul complet du facteur de forme, puisque la complexité des obstacles est nécessairement inférieure.

Dans le second cas, il est possible de gérer le temps de calcul de manière plus fine en imposant une borne supérieure au temps total alloué à l'évaluation du facteur de forme sous la forme d'un *tampon*. On autorise alors les appels au calcul exact tant qu'il reste du temps dans le tampon.

3.6 Calcul et stockage de l'information de visibilité

Le principe de l'utilisation de l'information multi-échelles de visibilité est que les quantités nécessaires au calcul de BVE et \overline{F}_C sont pré-calculées une fois pour toutes pour chaque cluster de la hiérarchie des obstacles, pour pouvoir être utilisées dans n'importe quelle configuration de calcul de facteur de forme. La propriété essentielle de ces informations (comme par exemple la transmittance directionnelle $\overline{\tau}$ est qu'elles ne dépendent que de caractéristiques propres aux obstacles et non de la configuration d'un calcul particulier de facteur de forme. Nous discutons dans cette section les différentes questions relatives au calcul et au stockage de l'information multi-échelles de visibilité.

3.6.1 Calcul

Une manière naturelle de calculer de l'information multi-échelle de visibilité, comme par exemple la transmittance directionnelle $\overline{\tau}$, serait d'effectuer un calcul récursif, *i.e* calculer les informations pour un obstacle donné en fonction de celles calculées pour ses fils. Malheureusement les quantités calculées ne représentent rarement en elles-mêmes une information suffisante pour assurer leur propagation ascendante. L'obstacle rencontré est l'équivalent du problème de la corrélation entre les sous-clusters dans la combinaison des facteurs de forme.

A l'inverse, nous pourrions calculer directement toutes les quantités concernées de manière indépendante d'un niveau à l'autre de la hiérarchie. Un tel algorithme manquerait ainsi de tirer profit de la cohérence existant entre la visibilité des obstacles à différents niveaux.

Ce que nous proposons de faire est propager de façon ascendante dans la hiérarchie des obstacles une information temporaire plus complète de laquelle nous déduirons à chaque niveau les quantités nécessaires au calcul de BVE et \overline{F}_C . Une fois ce calcul réalisé, seules ces dernières quantités subsistent. L'information temporaire que nous utilisons est constituée

d'un échantillonnage de la transmittance des obstacles, qui peut-être calculé récursivement. Au niveau des feuilles de la hiérarchie, la valeur initiale de l'information temporaire est directement déduite de la géométrie.

D'autre part, à un niveau donné de la hiérarchie, nous ne nous intéressons qu'aux niveaux de détail représentatifs de la taille du cluster concerné car il est important de ne pas prendre en compte les variations trop rapides de la fonction de visibilité, qui sont dues à des objets de taille très inférieure à la taille du cluster du niveau courant. Bien qu'elles n'apportent que des modifications négligeables dans le calcul du transfert d'énergie, ces perturbations peuvent fausser les critères de raffinement, qui sont justement basés sur les variations de la fonction de visibilité. L'opération de filtrage désirée est directement produite par la précision de l'échantillonnage qui est la même quelle que soit la profondeur de l'obstacle traité.

Le coût de calcul de l'information multi-échelle de visibilité est donc proportionnel au nombre total de *sous-clusters* dans la hiérarchie des obstacles. Cette méthode récursive de "décoration" de la hiérarchie des obstacle est par ailleurs largement parallélisable.

3.6.2 Stockage

Comme nous l'avons dit plus haut, l'information multi-échelles de visibilité peut prendre la forme de constantes ou de distributions directionnelles. Alors que le stockage d'une ou plusieurs constantes par *cluster* a une incidence faible sur le coût mémoire de notre méthode, la représentation de plusieurs fonctions directionnelles pour chaque cluster peut s'avérer coûteuse.

Pour une efficacité optimale de notre algorithme de calcul du facteur de forme, la méthode de stockage que nous employons doit répondre aux conditions suivantes : l'information multi-échelle de visibilité doit (a) être représentée de façon à permettre un accès rapide lors du calcul des facteurs de forme et (b) occuper le moins de place possible en mémoire.

Ces deux conditions sont par nature opposées. En effet, la manière de représenter une fonction de façon à faciliter au mieux le calcul de ses valeurs est de l'échantillonner. C'est également la méthode la plus gourmande en place mémoire. A l'inverse, représenter une fonction directionnelle comme combinaison linéaire sur une base de fonctions simples, offre souvent un taux de compression très intéressant, mais ralentit considérablement l'accès à ses valeurs. En ce qui concerne les fonctions directionnelles, on utilise le plus souvent une base d'harmoniques sphériques [SDS95] (voir également l'annexe A), ou d'ondelettes sphériques [SS95].

L'expérience montre qu'autant en dimension deux qu'en dimension trois, les fonctions directionnelles relatives à la visibilité d'un groupe d'objets peuvent être extrêmement irrégulières. Elles se prêtent donc très mal à une représentation dans une base de fonctions comme les harmoniques sphériques, et tendent à provoquer de l'*effet de Gibbs* autour des discontinuités. En revanche les ondelettes sphériques de faible degré permettent une représentation assez fidèle de ce type de fonction. Pour les raisons d'efficacité citées plus haut, nous avons choisi dans notre implémentation en dimension deux, n'étant pas limité pas la place mémoire, d'utiliser une représentation échantillonnée selon un nombre variable de directions des différentes quantités directionnelles intervenant.

3.7 Conclusion

Nous avons présenté une méthode générale de contrôle de l'erreur dans le calcul du facteur de forme entre deux surfaces, basée sur le pré-calcul et le stockage d'informations de visibilité à tous les niveaux de la hiérarchie des obstacles qui séparent ces surfaces. Grâce à ces *informations multi-échelles de visibilité*, notre l'algorithme est capable de décider

automatiquement de la profondeur de l'exploration de la hiérarchie des sous-obstacles de manière à produire un résultat dont l'erreur est inférieure à un seuil fixé à l'avance.

Pour une implémentation particulière, il nécessite cependant la spécification d'une approximation de la visibilité à travers un obstacle donné ainsi que le calcul explicite de bornes d'erreur associées à l'utilisation de cette information, et enfin une méthode d'estimation de corrélation de la visibilité entre sous-obstacles, au sujet desquels nous sommes volontairement restés très généraux.

Nous présenterons au chapitre 4 l'implémentation de notre algorithme dans le cadre de la radiosité en dimension deux, ce qui nous permettra de prouver par l'exemple son efficacité. Nous discuterons ensuite au chapitre 5 de son application au cas tri-dimensionnel. Comme le laisse en effet prévoir la différence de complexité géométrique entre la deuxième et troisième dimension, chacun de ces cas nécessite une étude spécifique.

Chapitre 4

Approximation contrôlée des facteurs de forme en dimension deux

LE CHAPITRE 3 nous a permis d'exposer le principe général d'algorithmes de calcul contrôlé du facteur de forme entre deux surfaces, par approximation de la visibilité à travers les obstacles qui séparent ces surfaces. Dans le présent chapitre, nous décrivons l'application de ces algorithmes au calcul contrôlé des facteurs de forme dans le cadre de la radiosité en dimension deux. Nous allons donc tout d'abord décrire comment fournir une approximation du facteur de forme en approchant la visibilité à travers les obstacles, ainsi que le moyen d'obtenir des bornes sur l'erreur d'approximation. Nous définirons précisément l'information multi-échelles de visibilité permettant de calculer cette approximation ainsi que les bornes d'erreur associées, pour toute configuration de facteur de forme utilisant les obstacles pour lesquels ces informations auront été pré-calculées. Pour finir, nous traiterons de l'estimation de la corrélation entre les sous-obstacles, qui permet d'effectuer le raffinement du calcul tout en contrôlant la précision. Finalement, nous illustrerons notre méthode par des exemples.

Nous commencerons tout d'abord par une brève introduction à la radiosité en dimension deux en dégageant ses caractéristiques spécifiques.

4.1 La radiosité en dimension 2

La radiosité en dimension deux, a souvent servi de premier domaine d'expérimentation d'algorithmes de calcul de l'éclairage [Ort97, HW91b] : on remarque en effet que la répartition de l'énergie lumineuse et les relations géométriques entre les objets sont plus simples à visualiser, les méthodes brutales de calcul utilisées pour produire des solutions de référence sont généralement plus efficaces et les calculs de facteurs de forme (par exemple en utilisant une intégration par méthode de Monte-Carlo) convergent relativement plus vite. Ces raisons font que les algorithmes sont en moyenne plus faciles à mettre au point en dimension deux, et nous bénéficierons dans ce chapitre des divers avantages de cette approche.

Comme en dimension trois, nous verrons que le problème de calculer la radiosité en dimension deux conduit à une équation intégrale, dont on ne connaît par ailleurs de solution analytique que dans un nombre très restreint de cas simples. Plusieurs approches ont été développées pour résoudre le problème, utilisant par exemple une méthode de Monte-Carlo [VFGP93], des éléments finis [HW91b], ou encore une méthode de collocation [Atk]. Des études plus poussées concernant les discontinuités de divers ordres ont également

été menées [HW91b, Atk]. Enfin, des outils spécifiques ont été développés, comme par exemple le complexe de visibilité 2D [ORDP96].

On pourra remarquer que le problème du calcul de la radiosité en dimension deux est équivalent au problème tri-dimensionnel obtenu en prolongeant infiniment la géométrie le long de la normale au plan de l'espace plat, ce qui justifie dans une certaine mesure son étude en vue d'approximations grossières du calcul dans des situations tri-dimensionnelles particulières.

Nous commencerons notre étude par quelques définitions, avant d'entrer dans le vif du sujet.

4.1.1 Définitions

Nous définissons dans cette section quelques notations essentielles, intervenant dans la modélisation de l'équilibre des échanges lumineux en dimension deux.

(a) Les objets du monde plat

En dimension deux, les objets sont délimités par des courbes de dimension un. Dans le cadre d'une simulation informatique des interactions lumineuses nous nous limitons donc à des courbes polygonales bornées. Les objets du monde deux-dimensionnel sont donc des segments.

Par convention, chaque segment admet une face extérieure et une face intérieure, définies par l'orientation du segment. Dans une scène correctement constituée les faces intérieures ne reçoivent jamais de lumière, comme dans l'exemple indiqué sur la figure 4.1.

Enfin, nous modélisons par ailleurs les rayons lumineux par des droites du plan.

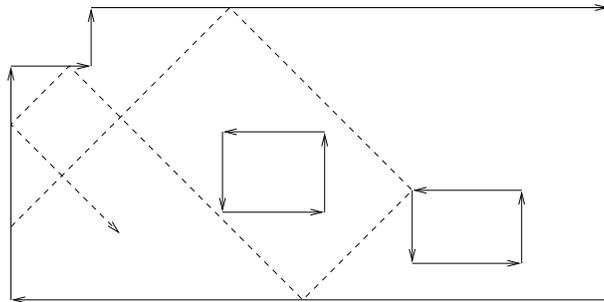


FIG. 4.1: Exemple de scène à deux dimensions. Les segments sont orientés. Seules les faces extérieures (i.e celles situées sur le côté droit) reçoivent de la lumière.

(b) Coordonnées polaires

Les droites du plan sont paramétrées, comme indiqué sur la figure 4.2, par un angle θ et une déviation ρ . L'équation cartésienne d'une droite $D(\rho, \theta)$ s'écrit donc

$$-x \sin \theta + y \cos \theta = \rho$$

Nous repérons dans ce contexte les directions par l'angle θ commun à tous leurs représentants.

(c) Mesures dans l'espace des droites

Comme nous l'avons annoncé au chapitre 3, il existe une unique mesure définie sur l'ensemble des droites du plan invariante par le groupe des déplacements du plan [Ort97].

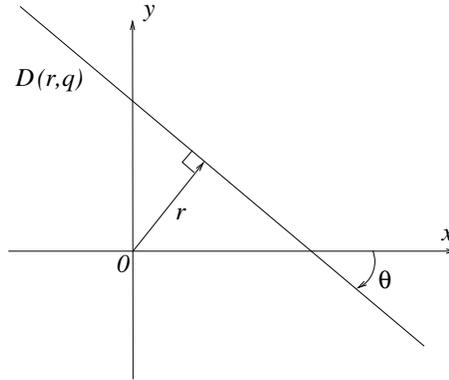


FIG. 4.2: Repérage d'une droite du plan par ses coordonnées polaires

Pour un ensemble X de droites du plan, cette mesure s'exprime naturellement en fonction de la paramétrisation des droites que nous utilisons comme :

$$\mu(X) = \int_{\{(p,\theta) | D(p,\theta) \in X\}} dp d\theta \quad (4.1)$$

En dimension deux, d'élégantes propriétés permettent de plus d'exprimer simplement la mesure de divers ensembles particuliers. Nous en rappelons deux :

Propriété 4 (Santalo [San76]) *La mesure de l'ensemble des droites qui rencontrent un convexe est égale au périmètre de ce convexe.*

En considérant un segment comme un convexe dont le périmètre vaut deux fois le diamètre, la mesure de l'ensemble des droites qui rencontrent un segment de longueur L vaut alors :

$$\mu(S) = 2L \quad (4.2)$$

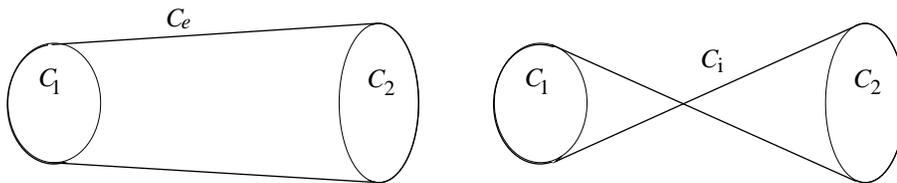


FIG. 4.3: Enveloppe convexe et courbe croisée de deux convexes disjoints.

Soient deux convexes bornés C_1 et C_2 disjoints. On définit alors les courbes C_e et C_i comme étant respectivement la frontière de l'enveloppe convexe de $C_1 \cup C_2$ et de la courbe croisée s'appuyant sur C_1 et C_2 , définie sur la figure 4.3. Nous notons également L_e et L_i les périmètres respectifs de ces courbes. On montre alors la propriété suivante :

Propriété 5 (Santalo [San76]) *La mesure de l'ensemble des droites qui rencontrent deux convexes disjoints C_1 et C_2 vaut :*

$$\mu(C_1, C_2) = L_i - L_e \quad (4.3)$$

Nous utiliserons ces propriétés dans le calcul du facteur de forme entre deux segments, notamment dans les cas de visibilité totale.

4.1.2 Équation d'équilibre

En dimension deux, dans le cadre de sources et réflecteurs idéalement diffus, la radiosité est définie comme la puissance lumineuse par unité de longueur. Elle s'exprime donc en *Watts par mètre* (noté $W.m^{-1}$). L'équation d'équilibre des échanges lumineux peut être établie de manière tout à fait similaire à la démonstration que nous en avons fait en dimension trois, au chapitre 1 (voir par exemple [Hec91]). Cette équation s'écrit :

$$b(x) = e(x) + \rho(x) \int_S \frac{\cos \varphi \cos \psi}{2r} v(x, y) dy$$

Les angles φ et ψ ainsi que la distance r sont ceux définis sur la figure 4.4. Le domaine d'intégration S représente l'ensemble des segments qui composent la scène.

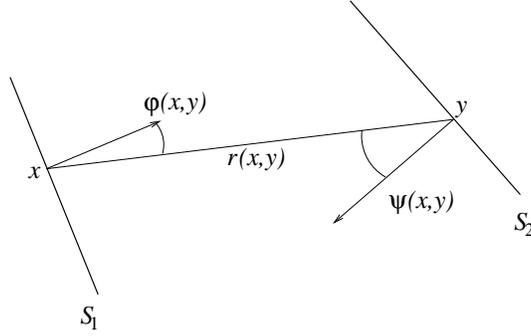


FIG. 4.4: Configuration générale de facteur de forme en dimension deux.

De même qu'en dimension trois, la discrétisation de l'équation de radiosité s'effectue par projection de la solution sur un espace vectoriel ayant pour base des fonctions polynomiales par morceaux. Dans le cas de fonctions constantes par morceaux, nous obtenons par la méthode décrite en 1.1.3 :

$$\forall i \quad B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

où les facteurs de forme F_{ij} sont définis par :

$$F_{ij} = \frac{1}{2L_i} \int_{S_i} \int_{S_j} \frac{\cos \varphi \cos \psi}{r} v(x, y) dx dy \quad (4.4)$$

Comme dans le cas des facteurs de forme en trois dimensions, le noyau de la radiosité est le terme :

$$k(x, y) = \frac{\cos \varphi \cos \psi}{r} v(x, y)$$

Il est considéré comme nul lorsque l'un des points x ou y est situé du côté de la face intérieure du segment sur lequel se trouve l'autre point.

4.1.3 Calcul des facteurs de forme

Nous allons décrire dans cette section deux expressions du facteur de forme que nous utiliserons par la suite. La première permet de calculer le facteur de forme dans les situations de visibilité totale entre deux segments. La seconde est une expression en coordonnées polaires de l'intégrale (4.4). Dans les deux cas, nous supposons que les segments sont en situation de *visibilité totale* i.e qu'aucun des deux segments ne coupe la droite supportant l'autre, sauf éventuellement au niveau de ses extrémités. Deux segments ayant un point commun partagent donc nécessairement une extrémité, et toute droite coupant les deux segments appartient au domaine d'intégration dans le calcul du facteur de forme.

Le facteur de forme entre les deux segments S_1 et S_2 s'exprime comme

$$F_{12} = \frac{1}{L_1} \int_{x \in S_1} \int_{y \in S_2} \frac{\cos \varphi \cos \psi}{2r} dx dy$$

Où φ et ψ sont les angles d'incidence de la droite (x, y) sur les segments, et r la distance entre ces points (C.f. Fig. 4.4).

(a) String rule

Supposons que deux segments S_1 et S_2 ne sont séparés par aucun obstacle. Nous reprenons l'expression du facteur de forme entre S_1 et S_2 en termes de mesures de droites définie à la section 3.1.1. En notant $\mu(S_1)$ (resp. $\mu(S_1, S_2)$) la mesure de l'ensemble des droites qui rencontrent S_1 (resp. S_1 et S_2), nous avons :

$$F_{12} = \frac{\mu(S_1, S_2)}{\mu(S_1)}$$

Le dénominateur s'exprime en utilisant (4.2), et le numérateur grâce à (4.3). Nous pouvons donc écrire que :

$$F_{12} = \frac{L_i - L_e}{2L_1}$$

En exprimant L_i et L_e en utilisant les distances entre les sommets, comme défini sur la figure 4.5.a, nous obtenons :

$$F_{12} = \frac{L_5 + L_6 - (L_3 + L_4)}{2L_1}$$

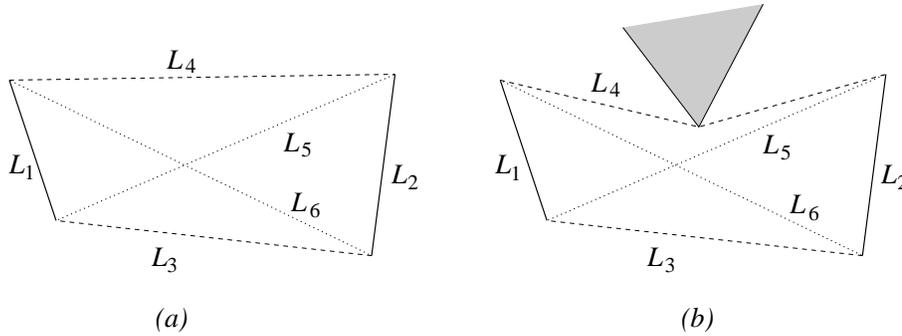


FIG. 4.5: Définition des courbes croisées et non croisées reliant deux segments.

Cette formulation du facteur de forme est connue sous le nom de *string rule*. Nous l'utiliserons pour calculer le facteur de forme entre deux segments dans les cas de visibilité totale. Cette formule fonctionne pour tout couple de convexes et s'étend par ailleurs à certains cas d'occlusion comme celui présenté sur la figure 4.5.b avec les notations qui y sont présentées [Ort97], mais nous n'en ferons pas l'usage. Par ailleurs, aucune expression similaire ne permet de calculer le facteur de forme entre deux convexes en dimension trois.

(b) Expression en coordonnées polaires

En utilisant l'équation (4.1), nous pouvons exprimer le facteur de forme en fonction de la transmittance τ de l'obstacle. Nous avons en effet :

$$\mu(S_1, S_2) - \mu(S_1, S_2, C) = \int_{\rho} \int_{\theta} \tau(\rho, \theta) d\rho d\theta$$

Le facteur de forme s'écrit donc en coordonnées polaires :

$$F_{12} = \frac{1}{2L_1} \int_{\rho} \int_{\theta} \tau(\rho, \theta) d\rho d\theta \quad (4.5)$$

Le domaine d'intégration est une région bornée du plan (θ, ρ) délimitée par quatre arcs de sinuséide. On peut donc le plonger dans un quadrilatère aux cotés parallèles aux axes :

$$\begin{aligned} \theta_{min} &\leq \theta \leq \theta_{max} \\ \rho_{min} &\leq \rho \leq \rho_{max} \end{aligned}$$

4.2 Approximations des facteurs de forme

Nous expliquons dans cette section deux méthodes d'approximation de la visibilité conduisant à des approximations du facteur de forme, et obtenons ainsi deux formulations différentes d'information multi-échelles de visibilité répondant aux caractéristiques définies au chapitre 3. Dans [Sol95] nous avons opéré en proposant une approximation du facteur de forme basée sur l'utilisation de la transmittance directionnelle, puis majoré la différence entre la valeur approchée et la valeur exacte en utilisant des normes usuelles. Après avoir brièvement rappelé ce résultat, nous allons décrire une méthode d'approximation beaucoup plus efficace, basée sur l'encadrement du facteur de forme puis en proposant comme approximation la valeur médiane des deux bornes.

Nous commencerons cependant par rappeler le résultat obtenu dans [Sol95], avant d'expliquer la seconde méthode.

4.2.1 Bornes basées sur l'emploi de normes usuelles

Dans [Sol95] nous avons proposé d'approcher le facteur de forme F_{12} entre deux segments S_1 et S_2 par le produit du facteur de forme sans occlusion entre les segments et la transmittance directionnelle $\bar{\tau}$ de l'obstacle, calculée dans une direction θ_{12} représentative des droites joignant les deux segments :

$$F_{12} \# \bar{\tau}F_0$$

Nous avons alors :

$$\|F_{12} - \bar{\tau}(\theta_{12})F_0\| \leq \frac{2\delta_{\theta}\sqrt{\Delta_{\rho}}}{L_1 r_{min}} [R_1(\theta_{12}) + R_2(\theta_{12})] \quad (4.6)$$

avec :

$$R_1(\theta) = \sup_{\theta' \in [\theta - \delta_{\theta}, \theta + \delta_{\theta}]} \|\tau(\cdot, \theta') - \tau(\cdot, \theta)\|_{L_2}$$

$$R_2(\theta) = \|\tau(\cdot, \theta) - \bar{\tau}(\theta)\|_{L_2}$$

Les différentes variables ont la signification suivante :

L_1	longueur de S_1
r_{min}	distance entre S_1 et S_2
θ_{12}	milieu de l'intervalle d'intégration en θ
δ_{θ}	demi-diamètre de l'intervalle d'intégration en θ
Δ_{ρ}	Diamètre de l'intervalle d'intégration en ρ pour $\theta = \theta_{12}$

Les quantités R_1 et R_2 dépendent uniquement de l'obstacle alors que θ_{12} , r_{min} , δ_θ , Δ_ρ et L_1 traduisent l'orientation relative des segments et dépendent donc de la configuration.

Avec ce choix d'approximation du facteur de forme, l'information multi-échelle de visibilité est alors constituée des fonctions directionnelles $\bar{\tau}$, R_1 , et R_2 , qui suffisent à calculer pour toute configuration une approximation du facteur de forme ainsi qu'une borne sur l'erreur.

A titre d'exemple, les quantités $R_1(\theta)$ et $R_2(\theta)$ sont représentées sur la figure 4.6 pour un obstacle particulier.

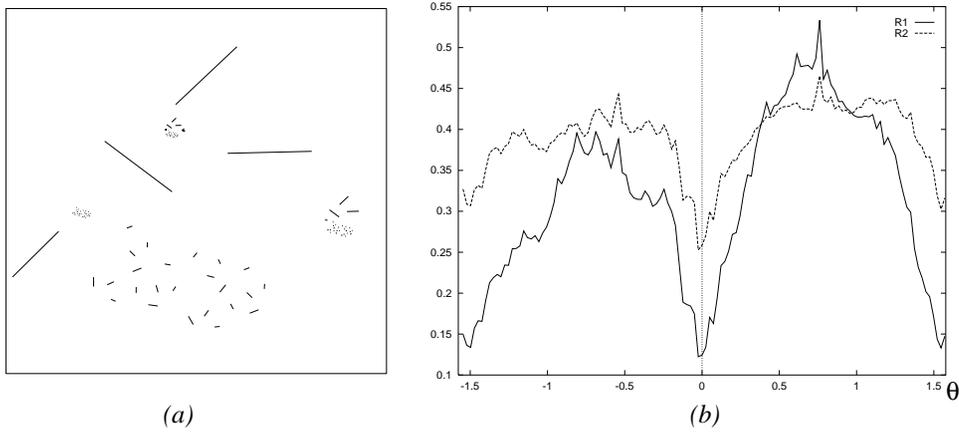


FIG. 4.6: Exemple d'obstacle (à gauche), accompagné des quantités $R_1(\theta)$ et $R_2(\theta)$ (à droite), pour $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

4.2.2 Bornes basées sur la géométrie des obstacles

Nous allons proposer maintenant une approche différente d'approximation du facteur de forme, basée sur des caractéristiques morphologiques des obstacles. Supposons tout d'abord que le domaine d'intégration ne contient que des droites qui coupent la boîte englobante de l'obstacle, comme sur la figure 4.7.a.

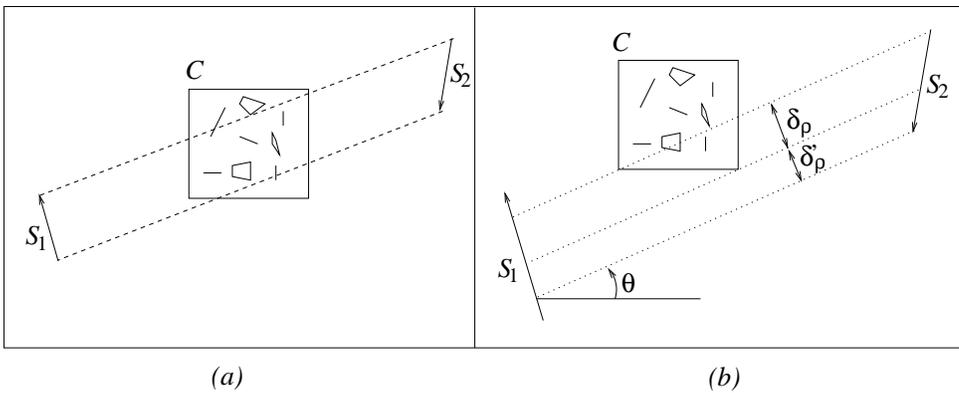


FIG. 4.7: (a) Configuration de type 'intersection totale'. (b) Configuration de type 'intersection partielle'.

L'expression (4.5) signifie que pour chaque valeur de θ , on mesure la longueur $\Delta(\theta)$ de la projection orthogonale de l'obstacle dans la direction θ (i.e la somme des longueurs

des composantes connexes des projections orthogonales de ses éléments) restreinte à l'intersection des projections des deux segments S_1 et S_2 dans la direction θ :

$$\Delta(\theta) = \int_{\rho_{min}(\theta)}^{\rho_{max}(\theta)} \tau(\rho, \theta)$$

La projection de l'obstacle dans une direction θ peut être représentée par sa fonction caractéristique f . Notons alors :

- δ_{min} le diamètre du plus petit intervalle sur lequel f vaut 1 ;
- δ_{max} le diamètre du plus grand intervalle sur lequel f vaut 1 ;
- $\tilde{\delta}_{min}$ le diamètre du plus petit intervalle sur lequel f vaut 0 ;
- $\tilde{\delta}_{max}$ le diamètre du plus grand intervalle sur lequel f vaut 0.

Nous pouvons alors majorer $\Delta(\theta)$ en remplissant au maximum l'intervalle $[\rho_{min}(\theta), \rho_{max}(\theta)]$ avec des intervalles de longueur δ_{max} séparés par des intervalles de longueur $\tilde{\delta}_{min}$; un majorant de $\Delta(\theta)$ est alors la somme des longueurs des intervalles de diamètre δ_{max} que l'on a pu utiliser.

De même, un minorant de $\Delta(\theta)$ est obtenu de manière duale à partir de $\tilde{\delta}_{max}$ et δ_{min} . Les bornes $\Delta_{max}(\theta)$ et $\Delta_{min}(\theta)$ s'écrivent alors :

$$\Delta_{min}(\theta) = \tilde{\delta}_{max} E(n) + \min(D(n)(\delta_{min} + \tilde{\delta}_{max}), \tilde{\delta}_{max})$$

$$\Delta_{max}(\theta) = \delta_{max} E(p) + \min(D(p)(\tilde{\delta}_{min} + \delta_{max}), \delta_{max})$$

avec

$$n = \frac{\rho_{max} - \rho_{min}}{\delta_{min} + \tilde{\delta}_{max}}$$

$$p = \frac{\rho_{max} - \rho_{min}}{\tilde{\delta}_{min} + \delta_{max}}$$

Dans l'expression précédente, les quantités $E(x)$ et $D(x)$ désignent respectivement la partie entière et la partie décimale de x . Nous avons donc

$$\frac{1}{2L_1} \int_{\theta} \Delta_{min}(\theta) \leq F_{12} \leq \frac{1}{2L_1} \int_{\theta} \Delta_{max}(\theta)$$

N.B : Il est possible de borner $\Delta_{min}(\theta)$ et $\Delta_{max}(\theta)$ indépendamment de θ . Cela évite une intégration pour calculer les bornes sur le facteur de forme, mais c'est moins précis.

Cas où l'obstacle ne recouvre pas toute la bande d'intégration

Nous nous intéressons maintenant au cadre plus général dans lequel les droites intervenant dans le calcul du facteur de forme ne rencontrent pas toutes la boîte englobante de l'obstacle (C.f figure 4.7.b). Nous notons alors $\delta_p(\theta)$ et $\delta'_p(\theta)$ les diamètres des intervalles indiqués sur la figure 4.7.b, correspondant à l'écart maximum des déviations des droites qui rencontrent l'obstacle ou pas. Pour exprimer de nouvelles bornes en fonction de celles obtenues au cas précédent, il suffit d'écrire que :

$$\Delta'_{min}(\theta) = \Delta_{min}(\theta)$$

$$\Delta'_{max}(\theta) = \delta_p(\theta) - \Delta_{max}(\theta, \delta'_p(\theta))$$

Lorsque le cluster est totalement inclus dans la bande d'intégration ces bornes restent valides si $\delta'_p(\theta)$ désigne la somme des largeurs des intervalles de part et d'autre du cluster.

Choix d'une approximation du facteur de forme

Disposant d'une borne inférieure F_{min} et supérieure F_{max} du facteur de forme, nous choisissons comme approximation du facteur de forme l'expression

$$\bar{F}_C = (F_{max} + F_{min})/2$$

L'erreur d'approximation est alors majorée par :

$$\epsilon_{max} = \frac{F_{max} - F_{min}}{2}$$

Et nous choisissons :

$$\text{BVE}(S_1, S_2, C) = \frac{F_{max} - F_{min}}{2}$$

L'information multi-échelles de visibilité est donc constituée au moins des quatre fonctions δ_{min} , δ_{max} , $\tilde{\delta}_{max}$ et $\tilde{\delta}_{min}$.

Cette méthode d'approximation a l'avantage de prendre implicitement en compte le niveau de détails des objets et donc d'être beaucoup plus cohérente. La figure 4.8 représente ces fonctions pour le groupe de segments servant d'obstacle dans la figure 4.6.

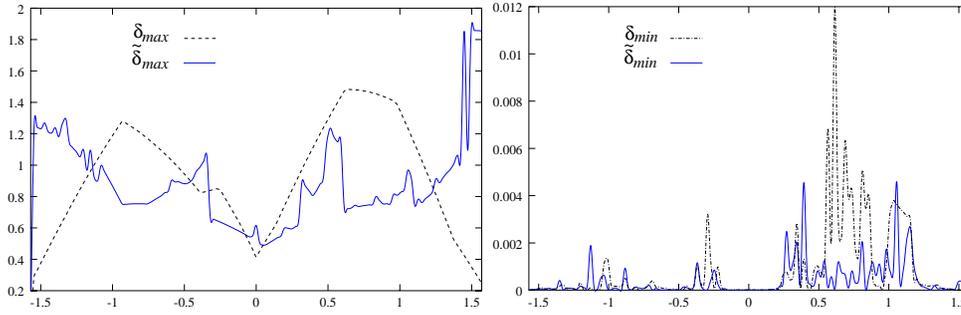


FIG. 4.8: Information multi-échelles de visibilité pour l'obstacle de la figure 4.6

Sur ces courbes on remarque tout d'abord l'écart important entre les ordres de grandeur des quantités δ_{min} et $\tilde{\delta}_{min}$ d'une part, et δ_{max} et $\tilde{\delta}_{max}$ d'autre part, venant du fait que l'obstacle de la figure 4.6 utilisé pour cette expérience est relativement complexe et contient surtout de nombreux niveaux d'échelle différents. La fonction d'erreur BVE utilisée par l'algorithme de calcul contrôlé du facteur de forme générera donc des bornes d'erreur très élevées dans les premiers appels récursifs, et forcera l'algorithme à subdiviser. Pour des obstacles plus simples, nous constatons en revanche, comme l'indique la figure 4.9 que les quantités sont du même ordre de grandeur, d'où un encadrement beaucoup plus serré de l'erreur d'approximation.

4.3 Estimation de la corrélation

Nous avons vu au chapitre 3 qu'une borne supérieure sur la corrélation de la visibilité entre les sous-obstacles est nécessaire pour contrôler l'erreur dans le calcul récursif du facteur de forme. Nous allons expliquer comment obtenir une telle borne.

Considérons un obstacle C ayant deux fils C_1 et C_2 . Le terme de corrélation χ défini en 3.4.1 représente la mesure des droites qui rencontrent à la fois S_1, S_2, C_1 et C_2 , notée $\mu(S_1, S_2, C_1, C_2)$.

Il est possible d'établir des bornes sur la corrélation entre les deux sous-clusters, comme nous l'avons fait pour le facteur de forme, en comblant des intervalles grâce aux fonctions

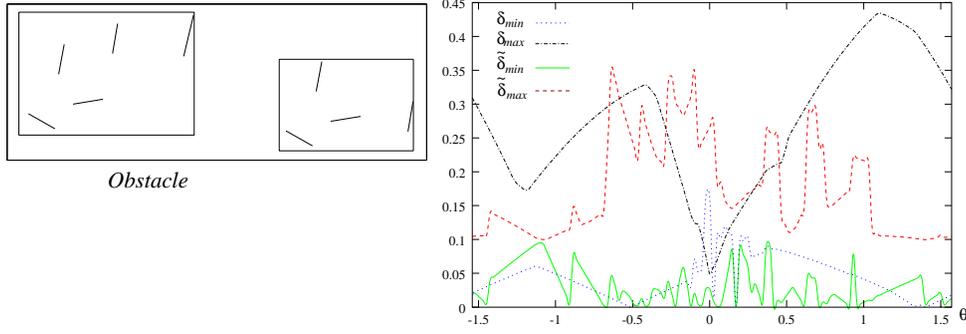


FIG. 4.9: *Information multi-échelles de visibilité pour un obstacle simple.*

δ_{min} , δ_{max} , $\tilde{\delta}_{min}$ et $\tilde{\delta}_{max}$. Alors que pour le facteur de forme, cette méthode produit des bornes très intéressantes, les bornes obtenues concernant la corrélation sont habituellement trop larges. Nous procédons donc autrement. Notons pour cela ω la *fonction d'opacité* d'un cluster :

$$\omega(\rho, \theta) = 1 - \tau(\rho, \theta)$$

Nous avons alors :

$$\mu(S_1, S_2, C_1, C_2) = \int_{\theta} \int_{\rho} \underbrace{\omega_1(\rho, \theta) \omega_2(\rho, \theta)}_{R(\theta)} d\rho d\theta$$

Si nous supposons que les surfaces S_1 et S_2 sont suffisamment loin l'une de l'autre, nous pouvons alors considérer le terme $R(\theta)$ comme constant sur l'intervalle d'intégration en θ , et représentable par sa valeur $R(\theta_{12})$ au centre de l'intervalle. Si l'on note δ_{θ} le diamètre de cet intervalle, nous pouvons donc écrire :

$$\mu(S_1, S_2, C_1, C_2) = \delta_{\theta} \int_{\rho} \omega_1(\rho, \theta_{12}) \omega_2(\rho, \theta_{12}) d\rho$$

Il faut ensuite restreindre l'intervalle d'intégration en ρ au diamètre δ_{ρ} correspondant à $\theta = \theta_{12}$, *i.e* :

$$\mu(S_1, S_2, C_1, C_2) \approx \delta_{\theta} \delta_{\rho} \chi(\theta_{12}) \quad \text{avec} \quad \chi(\theta_{12}) = \frac{1}{\Delta_{\rho}(\theta_{12})} \int_{\rho} \omega_1(\rho, \theta) \omega_2(\rho, \theta) d\rho$$

Cette approximation est d'autant plus légitime que le diamètre de l'intégration en ρ est petit lorsque l'on s'éloigne de θ_{12} . Cette expression nous permet d'approcher χ en fonction de quantités connues. Nous pouvons en effet écrire que :

$$\begin{aligned} \bar{\tau}(\theta) &= \frac{1}{\Delta_{\rho}} \int_{\rho} \tau_1(\rho, \theta) \tau_2(\rho, \theta) d\rho \\ &= \frac{1}{\Delta_{\rho}} \int_{\rho} (1 - \omega_1(\rho, \theta))(1 - \omega_2(\rho, \theta)) d\rho \\ &= 1 - \frac{1}{\Delta_{\rho}} \int_{\rho} \omega_1(\rho, \theta) d\rho - \frac{1}{\Delta_{\rho}} \int_{\rho} \omega_2(\rho, \theta) d\rho + \underbrace{\frac{1}{\Delta_{\rho}} \int_{\rho} \omega_1(\rho, \theta) \omega_2(\rho, \theta) d\rho}_{\chi_{\theta}} \end{aligned}$$

Notons maintenant Δ_{ρ_1} et Δ_{ρ_2} les diamètres des clusters C_1 et C_2 dans la direction θ . Nous avons alors :

$$\begin{aligned}\bar{\tau}(\theta) &= 1 - \frac{1}{\Delta_p} \int_{\Delta_{p_1}} \omega_1(\rho, \theta) d\rho - \frac{1}{\Delta_p} \int_{\Delta_{p_2}} \omega_2(\rho, \theta) d\rho + \chi'_\theta \\ &= 1 + \chi'_\theta - \frac{1}{\Delta_p} [\Delta_{p_1}(1 - \bar{\tau}_1(\theta)) + \Delta_{p_2}(1 - \bar{\tau}_2(\theta))]\end{aligned}$$

D'après cette relation, nous pouvons estimer la corrélation $\chi(\theta)$ par :

$$\chi(\theta) \# \bar{\tau}(\theta) + \frac{1}{\Delta_p} [\Delta_{p_1}(1 - \bar{\tau}_1(\theta)) + \Delta_{p_2}(1 - \bar{\tau}_2(\theta))] - 1$$

Il faut donc ajouter à l'information multi-échelle de visibilité servant à calculer BVE et \bar{F} , l'information nécessaire pour calculer χ . Dans le cas de l'approximation de F par $\bar{\tau}(d_1/2)F_0$ présentée en 4.2.1, la quantité $\bar{\tau}$ faisant déjà partie des fonctions stockées, la corrélation peut être calculée sans sur-coût de stockage. En revanche, dans le cas de l'approximation présentée en 4.2.2, il faut rajouter $\bar{\tau}$ à l'information multi-échelle de visibilité, ou directement χ . L'information multi-échelle de visibilité est donc formée de 5 fonctions directionnelles par sous-obstacle.

4.4 Résultats

Nous présentons dans cette section un exemple de calcul contrôlé du facteur de forme utilisant l'information multi-échelles de visibilité introduite en 4.2.2. La figure 4.10 représente la configuration du calcul du facteur de forme entre S_1 et S_2 , en présence d'obstacles formant une hiérarchie de racine C .

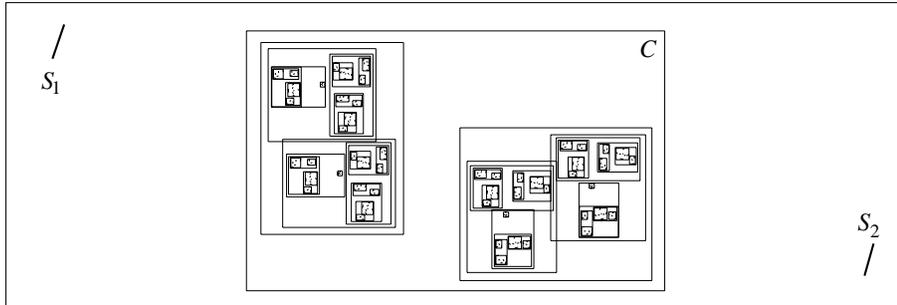


FIG. 4.10: Configuration expérimentale pour les courbes de la figure 4.11.

Nous avons pour cette expérience fait varier le seuil d'erreur ϵ_{max} de 0 à 0.0035, qui est la valeur au dessus de laquelle l'algorithme ne fait plus aucun raffinement. Comme on peut le constater sur la figure, la valeur calculée du facteur de forme reste contenue entre les bornes $F - \epsilon$ et $F + \epsilon$ imposées par le seuil d'erreur ϵ . Parallèlement, lorsque ϵ tend vers 0, le nombre de sous-obstacles visités augmente, ce qui traduit également l'augmentation de la profondeur des appels récursifs effectués par l'algorithme. Au fur et à mesure de ce raffinement, le temps de calcul augmente.

Les temps de calcul que nous avons mesurés dans cette expérience n'ont pas de signification précise. En effet, ceux-ci sont biaisés par l'utilisation du calcul exact dans les cas où le raffinement n'est plus possible. Ce calcul relativement lourd, multiplie globalement toutes les durées mesurées. Nous avons cependant jugé utile de l'utiliser pour permettre de tester jusqu'à une valeur arbitrairement petite de la précision, l'efficacité de notre algorithme. Dans le cadre d'une utilisation *a posteriori* de l'algorithme de contrôle de la précision (voir la section 3.5.(b)) ce sur-coût disparaît. Par ailleurs, dans cet exemple, le temps total de calcul de l'information multi-échelles de visibilité a été de 12.8sec.

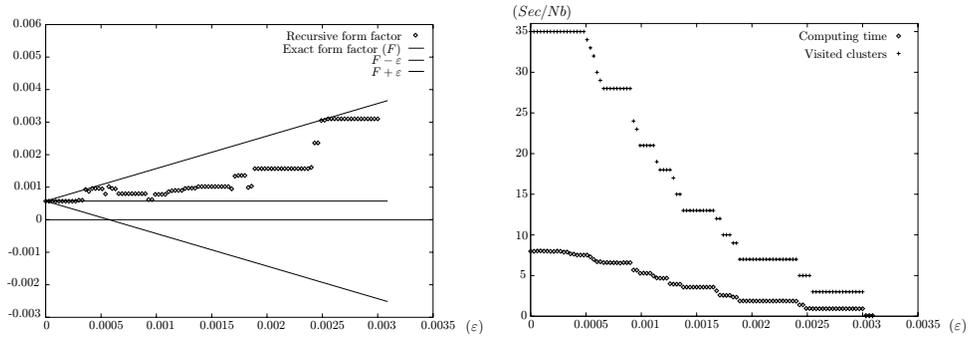


FIG. 4.11: Temps CPU de calcul du facteur de forme, et nombre total de sous-obstacles visités.

4.5 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre des algorithmes de calcul contrôlé du facteur de forme présentés au chapitre 3 dans le cadre de la radiosité en dimension deux. Nous avons vu deux approches différentes d'approximation du facteur de forme, ainsi que les quantités constituant l'information multi-échelles de visibilité correspondant à chaque choix. L'implémentation et l'expérimentation de ces techniques a prouvé la possibilité de contrôler l'erreur dans de calcul des facteurs de forme en dimension deux. La quantité de données correspondant à l'information multi-échelle de visibilité revêt notamment une taille raisonnable, et les bornes obtenues sont relativement fines.

Dans le chapitre suivant, nous discuterons de l'implémentation de ces mêmes techniques en dimension trois.

Chapitre 5

Approximation contrôlée des facteurs de forme en dimension trois

AU CHAPITRE 3, nous avons vu comment effectuer un calcul du facteur de forme entre deux surfaces, contrôlé par un seuil d'erreur fixé. Cette technique, basée sur une approximation hiérarchique de la visibilité, nécessite le pré-calcul et stockage d'une information à tous les niveaux de la hiérarchie des obstacles qui est ensuite utilisée pour l'estimation et l'encadrement de facteurs de forme impliquant ces obstacles. Au chapitre 4, nous avons décrit la mise en œuvre de cette technique dans le cadre de la radiosité en dimension deux, et montré qu'elle permet effectivement un contrôle de l'erreur. Dans le présent chapitre nous allons voir que, bien qu'il soit théoriquement possible d'appliquer les mêmes techniques en dimension trois, leur mise en œuvre les rend, en pratique, inintéressantes, notamment à cause de la difficulté de trouver des bornes suffisamment fines sur l'approximation de visibilité.

5.1 Encadrement du facteur de forme

Nous allons appliquer une méthode similaire à celle décrite en 4.2.2 pour estimer un encadrement du facteur de forme entre deux polygones A_1 et A_2 . Nous ne pouvons cependant pas, comme en dimension deux directement raisonner sur les aires des projections des objets constituant l'obstacle et de "l'espace vide" entre ces projections, pour combler l'intersection des projections des deux surfaces : pour ne pas obtenir un encadrement trivial nous serions en effet obligé de considérer le problème de la plus faible et plus forte densité d'un ensemble de polygones dans le plan, ce qui constitue un calcul très coûteux. Nous proposons donc une approche légèrement différente :

Nous avons vu au chapitre 3 (section 3.1.1) une expression du facteur de forme entre deux surfaces en termes de rapports de mesures d'ensembles de droites. Dans le cas de deux surfaces A_1 et A_2 , le facteur de forme vaut donc la mesure de l'ensemble des droites rencontrant A_1 et A_2 sans rencontrer d'obstacle, divisée par la mesure de l'ensemble des droites qui rencontrent A_1 . On montre par ailleurs [San76] que la mesure des droites rencontrant un convexe A de surface a vaut :

$$\mu(A) = \frac{1}{2\pi}a$$

D'autre part, en paramétrant les droites de l'espace par leur direction $\delta(\theta, \varphi)$ et la position s de leur intersection dans un plan orthogonale à δ , comme indiqué sur la figure 5.1, la mesure en dimension trois des droites contenues dans un ensemble X , vaut

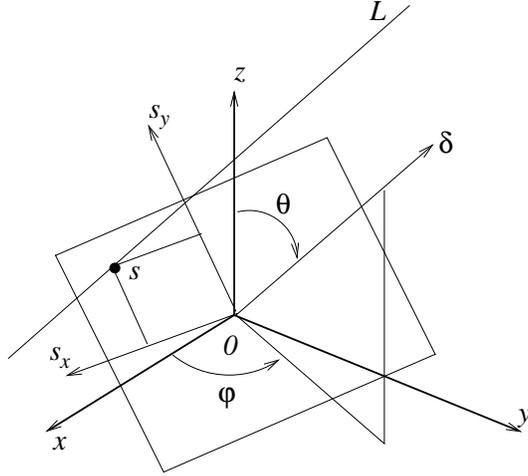


FIG. 5.1: Paramétrisation polaire des droites de l'espace par une direction δ repérée par deux angles (θ, φ) et une position s dans un plan orthogonal à cette direction.

exactement [San76] :

$$\mu(X) = \int_{\theta} \int_{\varphi} \int_x \int_y \sin\theta dx dy d\theta d\varphi$$

Nous pouvons alors exprimer le facteur de forme entre A_1 et A_2 en faisant intervenir la transmittance des obstacles qui les séparent :

$$F_{12} = \frac{1}{\pi A_1} \int_{\delta} \int_s \tau(L) d\delta$$

(Le module d'intégration $d\delta$ utilisé est celui correspondant aux coordonnées polaires des droites de l'espace, *i.e* $d\delta = \sin\theta d\theta d\varphi$)

Pour toute direction δ , nous notons $Proj_{\delta}(S)$ la projection d'un objet S sur un plan normal à δ , et posons $P(\delta) = Proj_{\delta}(A_1) \cap Proj_{\delta}(A_2)$, nous avons alors :

$$F_{12} = \frac{1}{\pi A_1} \int_{\delta} \underbrace{\int_{P(\delta)} Q_{\delta}(s) ds}_{P(\delta)} d\delta$$

où Q_{δ} désigne la fonction caractéristique de la projection de l'obstacle. Dans cette expression l'intégrale interne représente donc l'aire de la zone de $P(\delta)$ qui n'est pas masquée par la projection de l'obstacle comme l'illustre la figure 5.2.

Un encadrement du facteur de forme entre A_1 et A_2 pourrait alors être obtenu, selon une méthode similaire à celle utilisée en 4.2.2, à partir des valeurs extrémales $P_{min}(\delta)$ et $P_{max}(\delta)$ de $P(\delta)$ pour toute direction δ . Alors qu'il paraît impraticable de calculer ces valeurs extrémales pour tout ensemble $P(\delta)$, nous pouvons imaginer en disposer pour une forme particulière du domaine d'intégration. Si ce domaine est par exemple un un cercle de rayon r , notons $I_{min}(r, \delta)$, et $I_{max}(r, \delta)$ ces valeurs extrémales. Notons également $r_{inscrit}(\delta)$ et $r_{circonscriit}(\delta)$ les rayons des cercles inscrits et circonscriit à $P(\delta)$. Munis de cette information, nous pouvons alors borner le facteur de forme en utilisant l'encadrement :

$$\forall \delta \quad I_{min}(r_{inscrit}(\delta), \delta) \leq \int_{P(\delta)} Q_{\delta}(s) ds \leq I_{max}(r_{circonscriit}(\delta), \delta)$$

et donc :

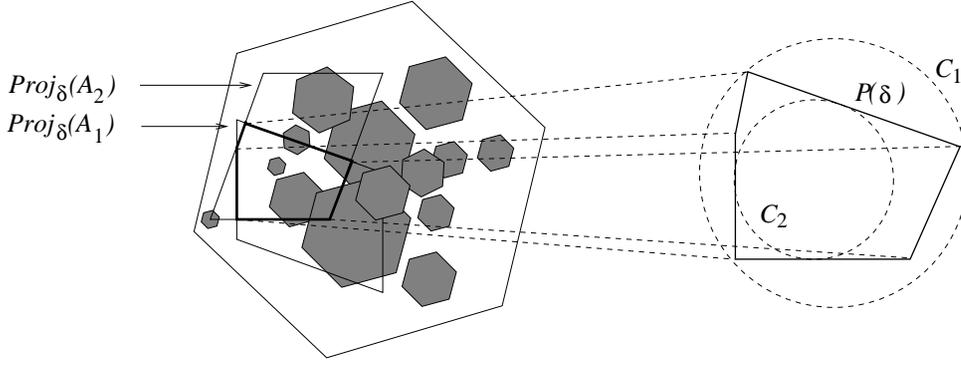


FIG. 5.2: Calcul de la zone d'intégration $P(\delta)$ par intersection des projections des surfaces A_1 et A_2 . A droite : encadrement de $P(\delta)$ par un cercle inscrit et un cercle circonscrit.

$$\underbrace{\frac{1}{\pi A_1} \int_{\delta \in D} I_{min}(r_{inscrit}) d\delta}_{F_{min}} \leq F_{12} \leq \underbrace{\frac{1}{\pi A_1} \int_{\delta \in D} I_{max}(r_{circonscriu}) d\delta}_{F_{max}}$$

Nous pouvons alors approcher F_{12} par :

$$\bar{F}_{12} = \frac{1}{2} (F_{max} + F_{min})$$

et une borne sur l'erreur d'approximation s'écrit alors :

$$|F_{12} - \bar{F}_{12}| \leq \frac{1}{2} (F_{max} - F_{min})$$

N.B : Une seconde possibilité serait, en disposant de la valeur moyenne $I_{mean}(r, \delta)$ de $P(\delta)$ pour tout cercle de rayon r , d'approcher F_{12} par :

$$\bar{F}_{12} = \frac{1}{\pi A_1} \int_{\delta \in D} I_{mean}(r_{mean}(\delta)) d\delta$$

où $r_{mean}(\delta)$ est le rayon d'un cercle "représentatif" de $P(\delta)$. Dans ce cas, l'encadrement de l'erreur n'est cependant plus immédiat.

Information multi-échelles de visibilité

Dans le cadre de ce que nous venons de proposer, l'information multi-échelle de visibilité nécessaire à l'estimation et à l'encadrement du facteur de forme consiste donc, en deux fonctions à trois paramètres (deux pour la direction, un troisième pour le rayon du cercle). Pour construire ces fonctions, il faut donc projeter l'obstacle selon un ensemble de directions et pour chacune d'entre elles mesurer dans l'image obtenue les valeurs minimum, maximum (*i.e.* $I_{min}(r, \delta)$, $I_{max}(r, \delta)$) et éventuellement $I_{mean}(r, \delta)$) de l'intégrale de la fonction caractéristique de la projection de l'obstacle multipliée par celle d'un disque de rayon r , pour des valeurs de r allant de 0 à la taille maximum possible. La figure 5.3 donne un exemple de ces fonctions à δ fixé.

L'estimation de la corrélation entre les sous-obstacles peut être réalisée en utilisant la méthode expliquée en 4.3, ce qui rajoute la transmittance directionnelle $\bar{\tau}$ aux fonctions directionnelles constituant l'information multi-échelle de visibilité.

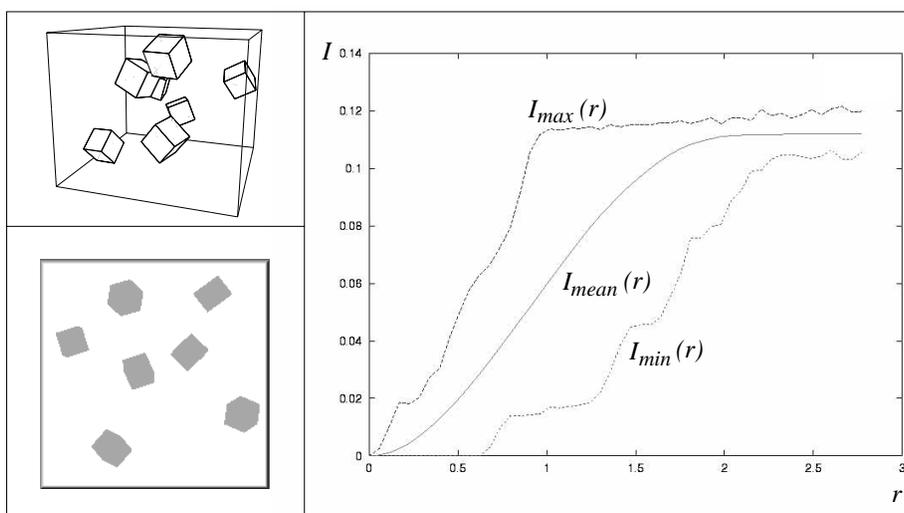


FIG. 5.3: Exemple d'analyse statistique d'un obstacle (en haut à gauche). L'image en bas à gauche représente la projection de l'obstacle dans une direction δ particulière. à droite : les valeurs $I_{min}(r)$, $I_{max}(r)$, ainsi que la valeur moyenne $I_{mean}(r)$ calculées pour 50 valeurs de r .

5.2 Discussion et conclusion

Bien que la solution que nous venons de proposer ne soit pas la seule solution envisageable, nous pensons qu'elle représente un exemple typique du compromis entre le temps de calcul et la quantité d'informations de visibilité stockées, nécessaires à l'implémentation en dimension trois des algorithmes décrits au chapitre 3. Basées sur cette expérience, nos conclusions sont les suivantes :

Les temps de calcul de l'information multi-échelle de visibilité sont tout d'abord assez longs, surtout si l'on considère qu'il faut les faire pour un nombre suffisamment représentatif de directions : les quantités I_{min} , I_{max} ou I_{mean} ne sont pas directement accessibles à partir de la géométrie tri-dimensionnelle des obstacles, essentiellement à cause de l'implication directe de la corrélation entre les objets qui s'y trouvent, tous niveaux d'échelle confondus. Pour les évaluer, il faut donc projeter l'obstacle et effectuer ces statistiques pour chaque direction. Les temps de calculs que nous avons mesurés sont globalement proportionnels à rn^2 où r est le nombre d'échantillons pour le rayon des cercle et n^2 le nombre de positions de son centre. En choisissant $n = r = 50$, le temps de calcul est par exemple de 30sec par direction sur une station SGI-O₂. Considérant qu'un choix de 20 directions est un strict minimum pour obtenir une représentation fidèle du contenu d'un obstacle comme celui montré sur la figure 5.3, et en négligeant le temps de calcul de la projection des obstacles (laquelle peut être réalisée très efficacement par rendu des dits obstacles en utilisant une projection orthogonale), le temps de calcul pour 1 obstacle est donc d'environ 10mn. Par ailleurs, une scène dans laquelle on désire effectuer une simulation de l'éclairage contient généralement plusieurs milliers d'obstacles emboîtés, ce qui rend le calcul de l'information multi-échelle de visibilité relativement coûteux.

La quantité d'informations à stocker est d'autre part très importante : en utilisant r valeurs d'échantillonnage pour I_{min} et I_{max} , il faut stocker $2 \times r$ fonctions directionnelles par cluster. A ce niveau intervient également un compromis entre d'une part, le temps de calcul

et la place mémoire et d'autre part l'accessibilité des informations : les méthodes de stockages à base d'harmoniques sphériques sont relativement économiques mais de mise en œuvre coûteuse (Les coefficients sont en effet calculés par intégration sur l'ensemble des directions ; voir aussi l'annexe A) et rendent l'accès aux valeurs des fonctions représentées assez coûteux. Stocker, d'autre part, un échantillonnage directionnel de $r = 50$ fonctions I_{min} et I_{max} pour un millier de clusters (en comptant 4 octets par valeur, ainsi que les directions correspondantes) demande environ 2.10^7 octets, *i.e* 20MB.

De plus, dans le cadre de l'obtention de bornes fiables sur l'erreur de visibilité, nous ne pouvons pas nous permettre de trop sous-échantillonner les directions car, à cause de la forte corrélation entre les objets dans un cluster donné, les fonctions directionnelles proposées peuvent avoir des variations assez brusques.

A ces remarques concernant le coût de mise en œuvre de l'information multi-échelle de visibilité en dimension trois, nous ajouterons que le coût du calcul des cercles inscrits et circonscrits d'un polygone n'est pas négligeable. Il faut de plus effectuer un traitement spécial lorsque le polygone d'intégration $P(\delta)$ n'est pas inclus dans la projection du cluster : on encadre alors la partie incluse et on complète avec la partie extérieure, comme nous l'avons fait dans le cas 2D.

En conclusion, nous pensons que, bien qu'elle soit théoriquement implémentable en dimension trois, la méthode que nous avons proposée au chapitre 3 n'est pas du tout adaptée à des simulations interactives. Le coût de sa mise en œuvre en dimension 3 est en effet trop important vis à vis de sa compétitivité face aux méthodes d'échantillonnage classiques.



Nous avons montré dans cette partie la faisabilité d'algorithmes de contrôle de l'erreur en radiosit  hi rarchiques, sur la base d'approximations multi- chelles de la visibilit . Ces algorithmes ouvrent la voie   des simulations quantitatives des  changes radiatifs dans des sc nes complexes. Dans les applications o  l'on attache une grande importance   la qualit  visuelle des r sultats, les mesures globales d'erreur que nous venons de pr senter sont en revanche de peu d'int r t. C'est par exemple le cas lors de l'utilisation de la simulation de l' clairage dans des applications interactives pour la r alit  virtuelle ou les jeux vid o. Dans la troisi me partie du pr sent document, nous allons nous int resser au contr le de l'erreur due aux approximations de la visibilit  selon des crit res de qualit  plus perceptifs. Dans ce contexte, l'impact visuel des approximations de la visibilit  se mesure essentiellement dans la qualit  des ombres. Nous avons donc cherch    produire des ombres artificielles mais exemptes d'artefacts visuels: nous avons tout d'abord tent , gr ce   une analyse fr quentielle de telles ombres, d'extraire des caract ristiques ou des informations permettant d'en g n rer   peu de frais. Nous pr senterons cette approche au chapitre 6. Nous avons  galement d velopp  une m thode permettant de calculer sans stockage suppl mentaire, des ombres de qualit  r glable, intrins quement exemptes d'artefacts, gr ce   l'utilisation de l'op ration de convolution. Nous en exposerons le principe au chapitre 7.

Troisième partie

Approche perceptive de l'erreur de visibilité

Chapitre 6

Génération d'ombres à partir de données fréquentielles

LE PROBLÈME abordé dans ce chapitre est la génération d'ombres *visuellement plausibles*, dues à obstacles complexes, à partir d'une faible quantité d'information concernant ces obstacles. Nous nous intéressons particulièrement à la question suivante : étant donné une source lumineuse, un obstacle et un récepteur, comment générer l'ombre de l'obstacle sur le récepteur, à partir d'un *petit échantillon* d'une ombre similaire (que nous appellerons *image de référence*), par exemple obtenue sur un récepteur plus petit, ou avec un obstacle plus simple dans d'autres conditions d'orientation ?

L'œil humain étant beaucoup plus sensible aux divers artefacts visuels contenus dans une image qu'à l'exactitude mathématique d'une ombre calculée, nous voulons tenter d'essayer d'extraire de l'image de référence des paramètres de type fréquentiels en quantité minimum (auto-corrélation, relation entre les différentes bandes d'octaves) pour permettre de synthétiser une image de même type.

Nous avons utilisé dans ce cadre deux approches différentes. La première a consisté à extraire de la transformée de Fourier de l'image de référence un certain nombre d'informations, comme par exemple son auto-corrélation, et de s'en servir pour générer une nouvelle image. Nous présenterons cette approche à la section 6.1. La seconde approche que nous avons expérimentée est directement issue d'une méthode proposée en 1997 par Jeremy De Bonet [De 97] utilisant une décomposition en pyramides de Gauss et de Laplace [BA83] de l'image de référence. Nous en parlerons à la section 6.2.

6.1 Conservation de l'auto-corrélation

L'*auto-corrélation* de l'image de référence I , qui constitue une mesure globale de la corrélation entre les différentes fréquences d'une image, nous a paru dans un premier temps constituer une information suffisante pour générer d'autres images du même type. Notre but étant de ne conserver qu'un minimum d'information, nous ne nous intéressons qu'aux fréquences de l'image de référence, qui contiennent l'information morphologique à conserver. Nous avons donc procédé ainsi :

La transformée de Fourier $F(I)$ de l'image de référence est décomposée en une matrice d'*auto-corrélation* C et une matrice de *phases* ϕ , dont les coefficients représentent respectivement le module et l'argument des coefficients de même abscisse et même ordonnée dans $F(I)$. Nous avons ainsi :

$$\forall i \forall j \quad F(I)_{i,j} = C_{i,j}(\cos(\phi_{i,j}) + i \sin(\phi_{i,j})) \quad (6.1)$$

Pour générer une nouvelle image, nous recopions la partie de la matrice d'auto-corrélation

de l'image de référence correspondant à l'intervalle des basses fréquences qui nous intéresse, dans une matrice C' que nous complétons par des zéros. Nous obtenons alors une nouvelle image en associant à la nouvelle matrice de corrélation une matrice de phases aléatoires, et en utilisant la formule (6.1). Nous calculons ensuite la transformée de Fourier inverse de cette image, en ne conservant du résultat obtenu que la partie réelle. Le processus complet est schématisé sur la figure 6.1).

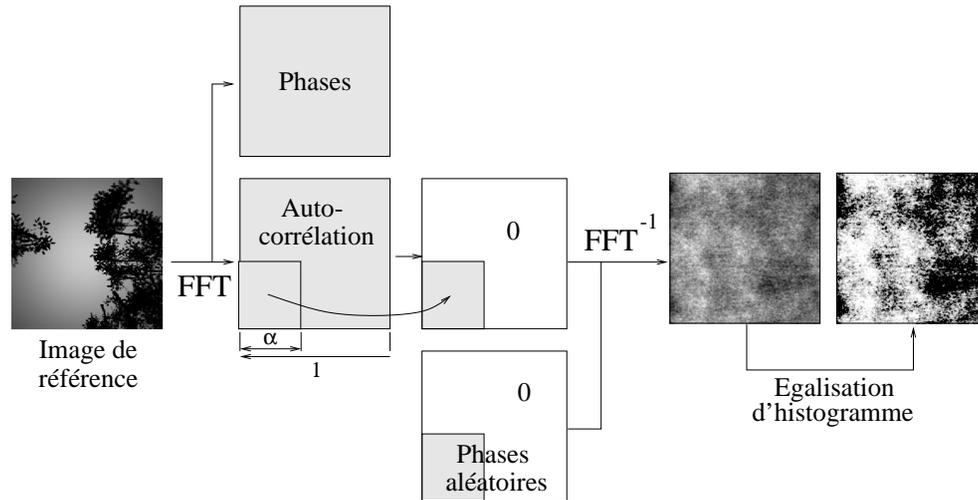


FIG. 6.1: Principe de la synthèse d'une texture d'illumination par transformée de Fourier : la transformée de Fourier de l'image de référence est scindée en une matrice de phases et une matrice d'auto-corrélation. On complète une partie de cette dernière par des phases aléatoires et le reste par des zéros. Après transformée de Fourier inverse, on effectue une égalisation d'histogramme.

La suppression des hautes fréquences permet de respecter dans l'image générée la taille minimale des détails de l'image de référence en éliminant les fréquences parasites comme par exemples celles dues à l'effet de Gibbs. Ayant choisi une matrice de phases aléatoire, de tels coefficients n'ont en effet plus de signification propre et polluent l'image synthétisée.

Nous représentons la quantité d'information conservée dans l'image de référence par le rapport α entre la taille de la partie de la matrice d'auto-corrélation effectivement utilisée et la taille de l'image entière. Pour reproduire des détails aussi fins que dans l'image de référence, il faut conserver toutes les informations correspondant à des longueurs d'onde supérieures au diamètre du plus petit détail de l'image. Soit T_{min} cette taille et soit S la taille de l'image de référence. Le ratio α doit donc vérifier l'inégalité :

$$\alpha \geq \frac{S}{2T_{min}}$$

Dans un premier temps, nous avons calculé la taille T_{min} manuellement.

Du fait que nous n'utilisons qu'une partie de l'information de l'image originale, l'image synthétisée ne contient pas la même répartition de niveaux de couleurs que l'image originale. Nous rajoutons donc en queue du traitement que nous venons de décrire, une *égalisation d'histogramme*. Il s'agit en effet d'une opération consistant à donner à l'image produite la même répartition du nombre de pixels par valeur d'intensité, que l'image de référence.

Selon le principe que nous venons d'exposer, l'information nécessaire à la génération d'une nouvelle image se limite aux valeurs des basses fréquences extraites de la matrice d'auto-corrélation de l'image de référence, *i.e* celles correspondant à des périodes supérieures à la taille des plus petit objets.

Nous présentons dans la section suivante les résultats qualitatifs que nous avons obtenus, avant de conclure.

6.1.1 Résultats

Nous avons testé la méthode décrite à la section précédente pour différents types d'images de référence. Nous en présentons deux, correspondant à des densités d'information sensiblement différentes. Ces images sont représentées sur la figure 6.2. L'image de droite est une ombre calculée à partir d'un modèle d'arbre.

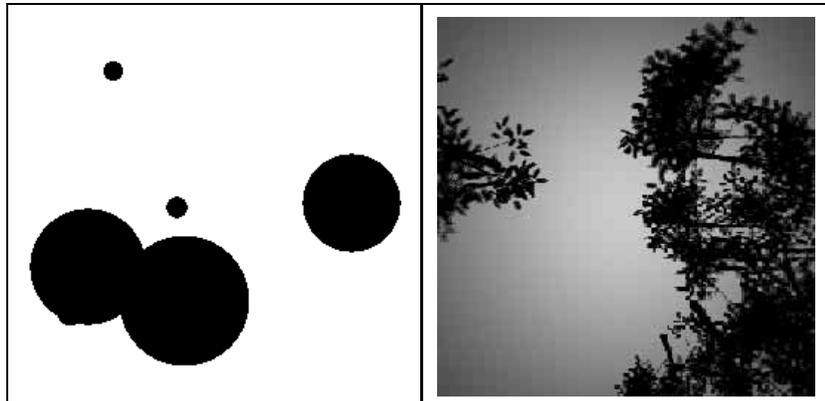


FIG. 6.2: Images de référence (Résolution 256×256)

La figure 6.3 présente les résultats obtenus pour les deux images de la figure 6.2, avec les valeurs de α correspondantes.

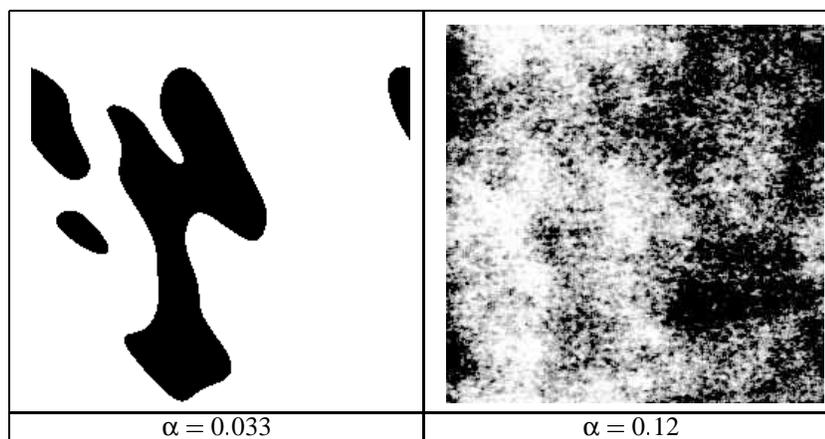


FIG. 6.3: Exemples de synthèses d'ombres à partir d'informations fréquentielles par la méthode décrite sur la figure 6.1. Le coefficient α indique le rapport de la taille de la sous-matrice d'auto-corrélation conservée dans l'image de référence.

Dans le cas des cercles, le plus petit détail est de 15 pixels, i.e $\alpha = 0.033$. Pour les arbres il est d'environ 4 pixels, d'où $\alpha = 0.12$. Ainsi, les résultats présentés sur la figure 6.3 correspondent-t-ils à $\alpha = 0.03$ pour l'image de gauche, ce qui revient à conserver 49 éléments de la matrice de corrélation. Pour l'image de droite, la valeur $\alpha = 0.10$ correspond à 25^2 éléments à conserver, i.e 625.

Ces expériences montrent qu'une quantité d'information relativement moins importante que le nombre de pixels de l'image de référence suffit à recréer une image *de même nature*. Nous remarquons cependant des différences essentielles entre images de référence et images générées, de localisation et de morphologie : les zones d'ombre et les zones éclairées ne sont tout d'abord pas placées aux mêmes endroits que dans l'image de référence. dans le cadre d'une reconstruction qualitative des ombres, nous considérons que cela ne constitue pas un obstacle majeur. Nous notons également des différences morphologiques : la forme des détails dans les ombres n'est pas totalement conservée par notre transformation. Nous aborderons ce problème à la section 6.2.

6.1.2 Conclusion

Dès qu'un obstacle susceptible de produire des ombres contient beaucoup de petits objets, nous sommes contraints de conserver énormément d'information pour arriver à générer des ombres correctes. La seule méthode envisageable est alors de négliger les petits éléments et synthétiser une ombre approximative lorsque nous nous en satisfaisons.

Un autre inconvénient majeur de la méthode que nous venons d'examiner est qu'elle ne permet pas de modifier la source de lumière. Une telle modification agit en effet de manière non triviale sur les coefficients de Fourier de l'image. On pourrait en revanche imaginer s'en servir pour générer la composante de visibilité uniquement et s'en servir comme *masque d'ombre* (ou *shadow mask*).

Cette méthode n'est d'autre part intéressante que dans le cadre d'obstacles isotropes du point de vue de l'aspect des ombres. Il serait en effet beaucoup trop coûteux de stocker une distribution directionnelle par coefficient de Fourier conservé.

Enfin, comme nous l'avons remarqué plus haut, la transformée de Fourier ne permet pas de conserver entièrement la *granularité* de l'image, *i.e* la cohérence entre les différentes bandes d'octaves. C'est pour cette raison que nous perdons la géométrie des formes. Cette observation confirme les travaux de Jeremy De Bonet, dont nous proposons d'examiner la méthode à la section suivante.

6.2 Échantillonnage par fréquence

Afin de capturer la cohérence entre les différentes bandes d'octaves dans l'image générée, De Bonet propose de contraindre l'échantillonnage des différentes bandes d'octaves de l'image synthétisée en fonction de la cohérence entre les bandes d'octaves de différents niveaux dans l'image de référence [De 97]. Nous décrivons dans cette section le principe général de cette technique et les résultats que nous avons obtenus lors de son application à la synthèse d'ombres.

6.2.1 Principe

La méthode proposée par J.De Bonet permettant de conserver la cohérence entre les différents niveaux de fréquence qui existent dans l'image de référence est la suivante :

L'image de référence est tout d'abord décomposée en pyramide de Laplace [BA83] dans une structure appelée *pyramide d'analyse*, servant à isoler l'information correspondant spécifiquement à chaque bande d'octaves. Pour calculer la pyramide de Laplace d'une image, on construit tout d'abord sa *pyramide de Gauss* en attribuant l'image de départ comme base de la pyramide, puis en calculant l'image correspondant à chaque niveau par moyennage de quatre pixels voisins dans l'étage d'en dessous, pour obtenir une image de dimensions moitié. Chaque niveau de la pyramide de Laplace est ensuite calculé comme la différence de deux niveaux successifs de la pyramide de Gauss, ramenés à la même taille d'image. Cette façon de procéder isole en quelque sorte l'information relative à la bande de fréquence correspondant à la résolution de chaque niveau de la pyramide.

On calcule ensuite, pour tous les niveaux de la pyramide de Gauss de l'image de référence, la réponse de chaque pixel à un ensemble de filtres : filtre de moyenne, de dérivée et de dérivée seconde dans les deux directions principales. On synthétise alors une nouvelle pyramide en commençant par simplement recopier les niveaux les plus hauts de la pyramide d'analyse, constitués d'un seul ou de quelques pixels, jusqu'à ce que la réponse aux filtres du dernier niveau créé puisse être calculée.

Les autres niveaux, situés sous ceux ayant été recopiés dans la pyramide d'analyse, sont construits de proche en proche, en recopiant dans le niveau équivalent de la pyramide d'analyse la couleur d'un des pixels dont les ancêtres ont une réponse suffisamment proche de celle des ancêtres du pixel courant dans la pyramide de synthèse. De tels candidats existent toujours, par construction.

L'image finale est alors obtenue en re-composant la pyramide de synthèse en une image. Il est possible par cette méthode d'obtenir une image plus grande que l'image de référence en recopiant plusieurs fois le sommet de la pyramide d'analyse dans la pyramide de synthèse. Le processus de synthèse de la nouvelle image est résumé sur la figure 6.4.

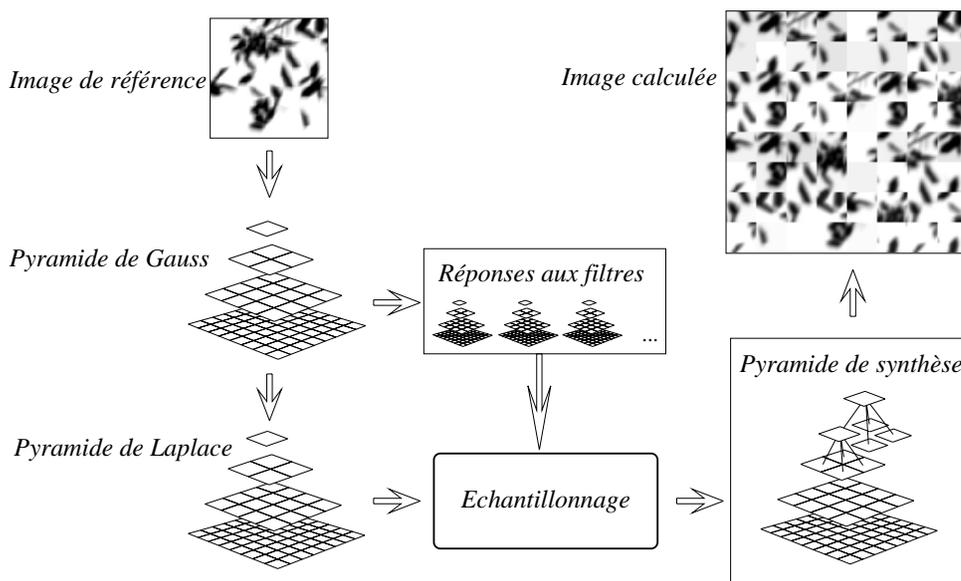


FIG. 6.4: Principe de la méthode d'échantillonnage fréquentiel : l'image de référence est décomposée en pyramide de Gauss, dans laquelle la réponse de chaque pixel à des filtres de dérivée directionnelle est calculée, puis en pyramide de Laplace. Une nouvelle pyramide de Laplace est confectionnée du haut vers de bas par sélection dans la pyramide de référence des pixels dont les ancêtres ont même réponse aux filtres utilisées. De cette nouvelle pyramide est ensuite déduite l'image synthétisée.

Cette méthode a l'avantage de traiter spécifiquement la corrélation entre les différentes bandes d'octaves d'une image, et a été utilisée pour générer des textures non répétitives, mais possédant la morphologie d'une texture de référence, plus petite. Nous avons conduit quelques expériences dans le but de tester son emploi pour la génération qualitative d'ombres.

6.2.2 Application à la génération des ombres

Nous avons implémenté et appliqué la méthode précédemment décrite pour des images de référence représentant des ombres réelles sans autre précaution, et en faisant varier le seuil en dessous duquel deux sites de la pyramide d'analyse sont considérés comme ayant des parents dont les réponses aux filtres sont indissociables.

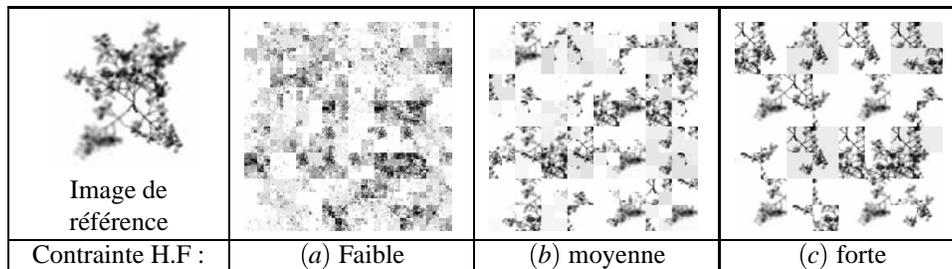
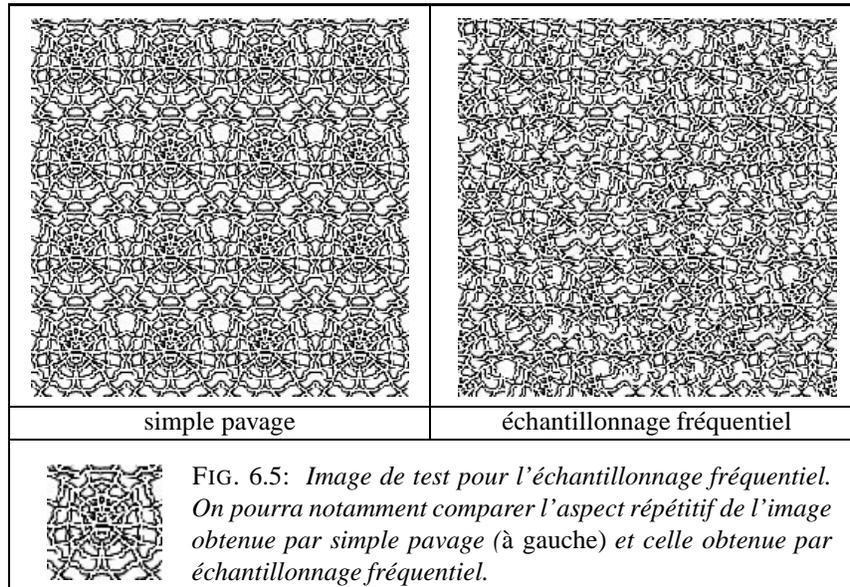


FIG. 6.6: Exemple d'application de l'échantillonnage fréquentiel à des images d'ombres réelles. De gauche à droite, la contrainte sur l'échantillonnage des hautes fréquences à été augmentée progressivement.

Le but de cette technique étant de générer des textures d'aspect non répétitif, nous ne pouvons pas trop contraindre l'échantillonnage des basses fréquences dans l'image synthétisée sans courir le risque d'obtenir la simple juxtaposition de n copies de l'image de référence. En revanche, pour conserver l'aspect de la texture de départ, les hautes fréquences, qui contiennent l'information, doivent être plus ou moins contraintes, selon la ressemblance désirée avec l'original. La figure 6.6 donne une idée du résultat généralement obtenu en fonction de différentes contraintes sur l'échantillonnage des hautes fréquences :

En imposant une faible contrainte sur l'échantillonnage des hautes fréquences, les valeurs des étages les plus bas de la pyramide de synthèse sont choisies de façon relativement indépendante de celles de leurs parents ($C.f$ figure 6.6.a). Nous retrouvons alors un comportement similaire à celui obtenu à la section précédente. En contraignant de manière plus forte l'échantillonnage des hautes fréquences ($C.f$ figure 6.6.b et 6.6.c), la morphologie des images générées se rapproche de celle de l'image de référence. En revanche, des discontinuités apparaissent entre les différents "blocs" issus de l'échantillonnage de basses fréquences. Ces discontinuités traduisent le défaut essentiel des ombres de références que nous avons choisies : les hautes fréquences ne sont pas indépendantes des basses fréquences.

Nous n'avons donc pas réussi, par cette méthode, à générer des ombres exemptes de tels artefacts, essentiellement parce qu'elle requiert de la part des échantillons de référence une propriété d'indépendance des hautes fréquences par rapport aux basses fréquences, que possède l'image nous ayant servi à tester la méthode (voire figure 6.2.1), mais qui est

rarement vérifiée par des ombres calculées de manière automatique.

6.3 Conclusions

Dans ce chapitre, nous avons implémenté et expérimenté deux techniques basées sur l'exploitation de différentes caractéristiques fréquentielles d'images de référence représentant des ombres, en vue de générer artificiellement des ombres plus étendues, d'aspect visuel plausible, et ayant les mêmes caractéristiques morphologiques. La première méthode, basée sur la conservation des plus basses fréquences de la matrice d'auto-corrélation de l'image de référence, nous a permis de vérifier que l'échantillonnage des différentes bandes d'octaves dans l'image synthétisée ne peut être réalisée indépendamment, sous peine de perdre le "grain" particulier à l'image de départ. La seconde méthode que nous avons expérimentée prend explicitement en compte la relation entre les différentes bandes d'octaves de l'image de référence pour échantillonner celles de l'image synthétisée. Malheureusement, cette méthode nécessite une hypothèse d'indépendance des hautes fréquences par rapport aux basses fréquences, qui n'est pas toujours vérifiée par les images de références qui nous concernent.

Nous n'avons donc pas réussi, par les deux méthodes présentées à atteindre l'objectif fixé au départ : obtenir des ombres exemptes d'artefacts, d'aspect général plausible, à partir d'un petit nombre d'informations fréquentielles. Nous pourrions également essayer d'autres approches, par exemple à base d'ondelettes, mais nous avons préféré orienter notre travail dans le sens de la technique que nous présentons au chapitre suivant.

Chapitre 7

Calcul d'ombres par convolution

IL EST FACILE de se convaincre en observant la figure 7.1 qu'à la différence des sources de lumière ponctuelles, qui produisent des *ombres dures*, les sources dites étendues sont à l'origine d'ombres progressives, ou *soft shadows*. Dans le cas de sources étendues, on ne peut en effet pas classer chaque point du récepteur comme voyant ou ne voyant pas la source. Il faut alors considérer la proportion de la source visible depuis ce point. Par conséquent, lorsqu'un obstacle produisant une ombre ne touche pas le récepteur, l'intensité lumineuse varie continuellement sur le récepteur.

Les ombres progressives jouent un rôle important dans la perception tridimensionnelle de la géométrie. Elles indiquent par leur vitesse de variation les distances relatives entre sources lumineuses, obstacles et récepteurs. Pour une même source étendue, l'ombre d'un obstacle est en effet d'autant plus *dure* que l'obstacle est proche du récepteur, ou que la source est loin de l'obstacle.

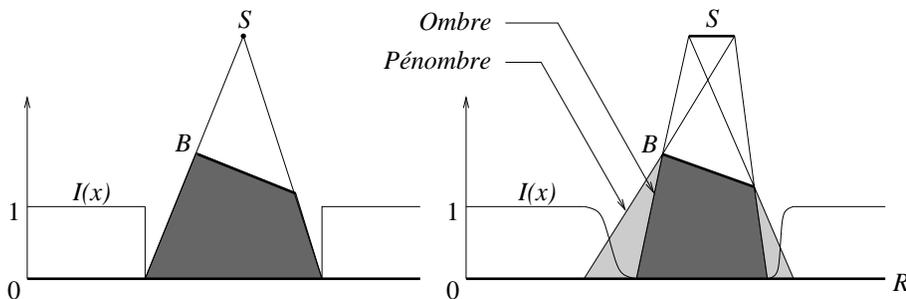


FIG. 7.1: Alors que les sources ponctuelles produisent des ombres dures (à gauche), les sources étendues forment des ombres progressives (à droite). On définit les zones d'ombre et de pénombre pour différencier les situations de visibilité des points du récepteur. Lorsque l'obstacle B ne touche pas le récepteur, l'intensité lumineuse $I(x)$ est une fonction continue de la position x sur le récepteur.

Le calcul des ombres progressives est malheureusement très coûteux : alors que les ombres dures peuvent être calculées par la simple projection depuis la source des obstacles sur le récepteur, le calcul des ombres progressives nécessite de déterminer exactement quelle est la proportion de la source visible depuis chaque point du récepteur.

De nombreuses méthodes de calcul des ombres dures, utilisant notamment les capacités graphiques des machines, ont été développées au cours des vingt dernières années [WPF90]. En revanche, nous verrons à la section 7.1 que le calcul des ombres progressives se heurte à différents problèmes : les méthodes analytiques comme la méthode du *maillage de discontinuité* souffrent de problèmes de robustesse et sont d'un coût prohibitif pour des scènes de complexité élevée. Les méthodes numériques, comme celles basées sur

l'échantillonnage de la source, produisent généralement des artefacts visuels désagréables lorsque l'on veut s'affranchir du coût trop important d'un échantillonnage très fin.

Nous allons présenter dans ce chapitre une méthode permettant de calculer efficacement les ombres progressives. Cette méthode, basée d'une part sur l'utilisation des capacités graphiques des machines et d'autre part sur l'application de la transformée de Fourier rapide, a l'avantage de produire des ombres intrinsèquement continues et donc exemptes d'artefacts visuels. Tout en étant approximative dans les configurations les plus générales notre méthode est assortie d'un algorithme permettant de régler la précision du résultat conformément à un critère d'erreur perceptif, au détriment du temps de calcul. Cette méthode a fait l'objet d'une publication à *Siggraph'98* [SS98b].

Avant de présenter le principe de la méthode de convolution (section 7.2), nous allons dresser un aperçu des diverses méthodes de calcul des ombres, et particulièrement des ombres progressives. Des questions plus techniques, notamment liées aux problèmes posés par la mise en œuvre pratique de notre méthode seront ensuite passées en revue (section 7.4), avant de traiter du contrôle de l'erreur proprement dit (Section 7.5).

7.1 État de l'art

De très nombreuses méthodes de calcul d'ombres ont été développées dans le domaine de la synthèse d'images. Nous nous contenterons de passer en revue les plus significatives pour positionner notre travail en nous focalisant sur les méthodes permettant de calculer des ombres progressives. Pour une approche plus systématique du domaine, le lecteur est invité à consulter l'état de l'art proposé par Woo *et al.* [WPF90].

7.1.1 Méthodes d'échantillonnage

La méthode du *lancer de rayon* effectue le calcul des ombres en traçant des rayons lumineux fictifs depuis chaque point visible de l'image calculée vers chaque source lumineuse [Whi80]. On sait alors que la source lumineuse éclaire le point concerné lorsque le rayon les reliant ne rencontre aucun autre objet dans la scène. Dans le cas de sources étendues, en lançant plusieurs rayons vers chaque source on peut déterminer approximativement la proportion de la source visible depuis ce point, ce qui permet de produire des ombres progressives [CPC84]. Cette méthode est néanmoins coûteuse puisqu'elle nécessite de nombreux tests d'intersection pour chaque rayon lancé avec tous les polygones de la scène.

Bien qu'elle soit difficilement adaptable au calcul des ombres progressives, la méthode du *shadow buffer* [Wil78] mérite d'être citée, comme exemple de méthode basée sur le calcul d'une image de la scène vue d'un point de la source. Les valeurs des profondeurs des points de cette image, lues dans le *z-buffer* sont utilisées comme valeurs de référence pour déterminer si un point est dans l'ombre : pour chaque point de la scène, on compare sa profondeur depuis la source à celle du pixel ayant même position dans l'image de référence. On sait alors si le point testé est dans l'ombre ou non. Il faut noter que dans sa version la plus simple, cette méthode souffre de sévères problèmes d'*aliassage* : un polygone trop incliné par rapport à la source voit certains de ses points abusivement masqués par ses voisins. L'utilisation d'une méthode de filtrage (ou *depth filtering* [RSC87]) permet néanmoins de résoudre ce problème.

7.1.2 Méthodes interactives

Diverses méthodes s'appuient sur la puissance graphique des machines pour produire des ombres à une vitesse interactive, en utilisant les cartes graphiques pour effectuer simultanément des calculs concernant des milliers de points.

Il est par exemple possible d'utiliser les fonctions de plaquage de texture (*texture mapping*) pour effectuer automatiquement la comparaison des profondeurs des pixels et ainsi déterminer s'ils se trouvent dans l'ombre [SKv⁺92]. Une telle méthode calcule donc automatiquement l'ombre d'une source ponctuelle au moment du rendu.

Beaucoup plus rares sont les méthodes interactives qui calculent des ombres progressives. L'une d'entre elles, présentée par Heckbert et Herf [HH97] consiste à calculer plusieurs images d'un même récepteur depuis des points distribués sur une source étendue. Chaque image représente alors ce récepteur, partiellement masqué par la projection des polygones (*i.e* l'ombre) placés entre lui et la source. En calculant efficacement la moyenne de ces images grâce à l'*accumulation buffer*, on obtient une image du récepteur sur lequel figure une approximation de l'ombre progressive issue de la source étendue. L'inconvénient majeur de cette méthode est qu'elle nécessite de calculer beaucoup d'images depuis la source pour que leur moyenne ne ressemble pas à une simple superposition d'ombres dures.

7.1.3 Utilisation de structures auxiliaires

Un certain nombre de structures construites à partir de la géométrie tri-dimensionnelle de la scène permettent d'effectuer rapidement des requêtes de visibilité. Ces structures sont basées sur la cohérence de la visibilité dans la scène.

Les volumes d'ombre (ou *shadow volumes*) sont des cônes tronqués dont le sommet est une source ponctuelle qui s'appuie sur les polygones de la scène. Il est facile de tester si un point est éclairé par la source située au sommet du cône en testant l'appartenance de ce point au volume d'ombre [Cro84]. L'usage d'*arbres BSP* permet d'effectuer de façon efficace les tests d'appartenance aux volumes les plus complexes [CF89]. Cette technique peut s'appliquer au calcul des ombres progressives en disposant plusieurs sources ponctuelles sur une source étendue donnée [BB84].

Certaines structures plus efficaces sont basées sur la catégorisation des rayons lumineux en termes d'événements de visibilité. Les surfaces réglées délimitant les différentes catégories sont le lieu géométrique des discontinuités dans les dérivées successives de l'illumination d'un polygone sur un autre. Le maillage de discontinuité [Hec92, LTG93, DF94] recense ainsi une partie ou toutes les discontinuités d'illumination à partir d'une source étendue et permet de conduire la discrétisation des méthodes d'illumination globale de façon à respecter ces discontinuités. Le *squelette de visibilité* [DDP97] effectue un recensement plus global de ces événements entre tous les objets de la scène. Il permet d'effectuer des requêtes ordinairement coûteuses en un temps très court : maillage de discontinuité entre deux polygones quelconques, polygone visible depuis un sommet donné, Bien que ces structures fournissent une information exacte de visibilité et ouvrent donc la voie à des images d'excellente qualité [DF94], leur construction est très délicate à implémenter et généralement assez sensible aux imprécisions numériques et aux dégénérescences.

7.1.4 Ombres et méthodes d'éclairage global

La méthode de radiosité vue au chapitre 1, permet de calculer les ombres progressives puisqu'elle prend implicitement en compte la visibilité entre les surfaces élémentaires échangeant de l'énergie au travers des facteurs de forme. Dans la plupart des implémentations, les facteurs de forme sont calculés en découplant le terme de visibilité du terme d'illumination directe, permettant ainsi la mise en œuvre d'un calcul spécifique de la visibilité [CG85, WEH89, HSA91]. Dans cette optique, Zatz [Zat93] propose de calculer pour un couple source-récepteur une carte d'ombre (*shadow map*) dans laquelle est stockée la position des ombres, ce qui permet une prise en compte beaucoup plus fidèle des ombres au moment du calcul des transferts d'énergie et plus réaliste lors de l'affichage de la solution.

De plus, dans le cas de scènes idéalement diffuses, la radiosité, aussi complexe soit-elle, peut être stockée au niveau des surfaces sous la forme de *textures* [HS93], ce qui permet

ensuite d'afficher le résultat du calcul très rapidement [MK94] mais aussi de l'utiliser pour d'autres transferts d'énergie [Hec90]. Ainsi Keller [Kel97] calcule la radiosité en rendant plusieurs fois la scène vue depuis différents points sur les sources de lumière, puis en moyennant les images.

7.1.5 Méthodes multi-échelles

Bien que les ombres soient d'une grande importance dans la perception tri-dimensionnelle de la géométrie, l'expérience montre que leur degré d'imprécision importe peu, de même qu'une simple tache sombre dessinée sous un objet permet de situer celui-ci dans l'espace [Wan92]. Cette idée est à l'origine de la méthode proposée par Sillion [SD95] qui produit des ombres à divers degrés d'approximation en sélectionnant dans la hiérarchie des objets faisant occlusion un niveau approprié de simplification des relations de visibilité sous la forme d'objets semi-transparents.

7.1.6 Position de notre travail

Notre but a été de construire un algorithme permettant de produire des ombres progressives à une vitesse interactive, mais en évitant les artefacts visuels des méthodes basées sur l'échantillonnage. Le critère d'interactivité ne nous permettant pas d'utiliser des structures complexes comme le maillage de discontinuité, nous avons opté pour une méthode calculant la visibilité à la volée, en utilisant de façon optimale les capacités des machines graphiques.

La méthode que nous allons présenter est basée sur l'opération de convolution entre des images bien choisies d'une source étendue et d'un obstacle produisant des ombres. Elle peut être considérée comme l'extension au cas général d'une méthode introduite par N.Max en 1991 [Max91] pour calculer l'éclairage du dôme celeste sur le sol, *i.e* dans le cas particulier d'une source située à l'infini.

7.2 Principe de la méthode de convolution

Nous présentons tout d'abord notre méthode de convolution dans un cas d'école où le terme exprimant la visibilité sur le récepteur s'écrit de façon exacte comme une convolution. La méthode est ensuite étendue au calcul des ombres dans le cas général (Section 7.2.2).

7.2.1 Cas d'école : objets plans et parallèles

Considérons une source polygonale S d'intensité uniforme E , un récepteur plan R et un obstacle également plan B . Supposons également que les plans dans lesquels ces trois objets sont respectivement inclus, sont parallèles. Nous notons alors d_1 et d_2 les distances entre ces plans, comme indiqué par la figure 7.2.

Utilisons les notations introduites au chapitre 1. L'irradiance en un point y du récepteur due à la source S s'écrit :

$$H(y) = \frac{E}{\pi} \int_S \frac{\cos \theta \cos \theta'}{r^2} v(x, y) dx$$

Comme nous l'avons remarqué au chapitre 2, il est légitime d'approcher cette expression en négligeant la corrélation entre la visibilité et le noyau de la radiosité :

$$H(y) \# E \underbrace{\int_S \frac{\cos \theta \cos \theta'}{\pi r^2} dx}_{F_S(y)} \underbrace{\int_S v(x, y) dx}_{V(y)} \quad (7.1)$$

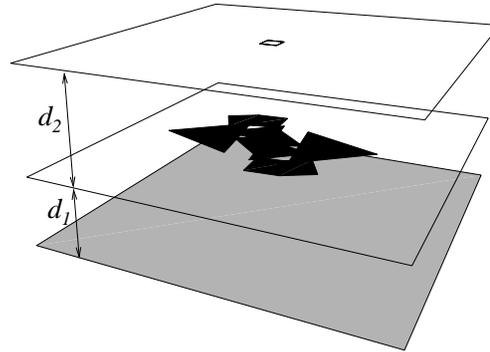


FIG. 7.2: Cas d'école dans lequel le terme exprimant la proportion de la source visible depuis chaque point du récepteur, s'exprime comme une convolution. Les trois éléments sont plans et parallèles. Les distances entre les plans sont notées d_1 et d_2

Dans cette approximation, $F_S(y)$ représente le facteur de forme de y vers la source S , et $V(y)$ l'aire de la source visible depuis y à travers l'obstacle B . Nous nous intéressons plus particulièrement au calcul de $V(y)$.

Calculer $V(y)$ revient à projeter l'obstacle B sur la source S depuis le point y et mesurer la portion de l'aire de la source qui n'est pas masquée. Les trois éléments étant parallèles, la projection de B sur S se translate dans le plan de la source, sans se déformer, lorsque y parcourt le récepteur. Cette propriété, qui découle simplement du théorème de Thalès, est illustrée en dimension 2 sur la figure 7.3.a. Grâce à cette propriété, nous allons pouvoir exprimer $V(y)$ comme une convolution.

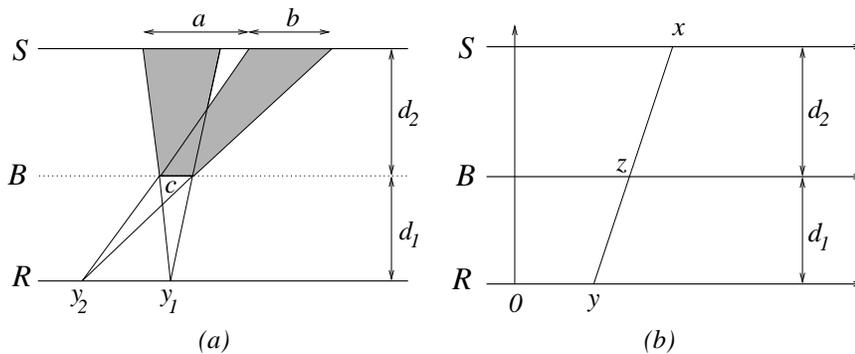


FIG. 7.3: (a) Dans le cas d'éléments parallèles, la projection d'un obstacle sur la source depuis un point du récepteur se translate avec ce point. Dans cet exemple, a indique la translation de l'ombre ($a = \frac{d_2}{d_1} |y_2 - y_1|$) et b est sa taille, indépendante de y (Nous avons en effet $b = \frac{d_1 + d_2}{d_1} c$). (b) Position du point d'intersection d'un rayon (x, y) avec le plan de l'obstacle.

Nous allons utiliser les fonctions suivantes, définies dans les plans respectifs de la source et de l'obstacle :

$$\begin{aligned} S(x) &= \begin{cases} 1 & \text{si } x \text{ est sur } S \\ 0 & \text{sinon.} \end{cases} \\ P(x) &= \begin{cases} 0 & \text{si } x \text{ est sur } B \\ 1 & \text{sinon.} \end{cases} \end{aligned}$$

Dans le plan de l'obstacle, le point d'intersection d'une droite joignant x sur S et y sur R , avec ce plan, a pour affixe z défini par (C.f figure 7.3.b) :

$$z(x, y) = \frac{d_1 x + d_2 y}{d_1 + d_2}$$

En utilisant P pour exprimer la visibilité entre x et y , nous avons alors :

$$V(y) = \int_S P\left(\frac{d_1 x + d_2 y}{d_1 + d_2}\right) dx$$

Faisant intervenir la fonction S , nous étendons cette intégrale à tout le plan :

$$\begin{aligned} V(y) &= \int_{\mathbb{R}^2} S(x) P\left(\frac{d_1 x + d_2 y}{d_1 + d_2}\right) dx \\ &= \left(\frac{d_2}{d_1}\right) \int_{\mathbb{R}^2} S\left(\frac{d_2}{d_1} x\right) P\left(\frac{d_2}{d_1 + d_2}(x + y)\right) dx \end{aligned}$$

Posons maintenant :

$$\alpha = \frac{d_1}{d_2} \quad s_\alpha(x) = S\left(-\frac{1}{\alpha}x\right) \quad p_{1+\alpha}(x) = P\left(\frac{1}{1+\alpha}x\right)$$

Nous avons alors

$$\begin{aligned} V(y) &= \frac{1}{\alpha^2} \int_{\mathbb{R}^2} s_\alpha(-x) p_{1+\alpha}(x + y) \\ &= \frac{1}{\alpha^2} \int_{\mathbb{R}^2} s_\alpha(x) p_{1+\alpha}(y - x) \\ &= \frac{1}{\alpha^2} (s_\alpha \star p_{1+\alpha})(y) \end{aligned} \tag{7.2}$$

Le terme de visibilité $V(y)$ peut donc être calculé comme une convolution (notée ' \star ') entre une image de la source et une image de l'obstacle mises à une échelle adéquate. La figure 7.4 montre le résultat de la convolution dans le cas d'école présenté sur la figure 7.2 pour deux valeurs différentes de α .

En reprenant la formulation de l'irradiance (7.1) et en notant $\rho(y)$ la réflectance sur le récepteur, la radiosité au point y issue de S s'écrit :

$$b(y) = \frac{\rho(y)E}{\alpha^2} F_S(y) (s_\alpha \star p_{1+\alpha})(y)$$

Nous proposons donc de rendre la radiosité sur le récepteur en utilisant l'image de convolution comme une *texture* que nous modulons par le facteur $\frac{\rho(y)E}{\alpha^2} F_S(y)$. La texture représente alors de l'information de visibilité et est utilisée comme masque d'ombre (*shadow mask*) sur le récepteur. Nous appellerons par la suite *texture de visibilité* l'image obtenue par convolution. La texture de visibilité, complétée des valeurs d'éclairage direct $\frac{\rho(y)E}{\alpha^2} F_S(y)$, sera désignée par le terme *texture de radiosité*, puisque l'ensemble représente de la radiosité sur le récepteur.

Comme nous le verrons dans la section suivante, l'expression (7.2) n'est valable que dans le cas particulier d'éléments plans et parallèles. Dans les autres configurations, elle constitue cependant une approximation très intéressante du terme de visibilité. Nous allons maintenant discuter de son utilisation dans un cadre plus général.

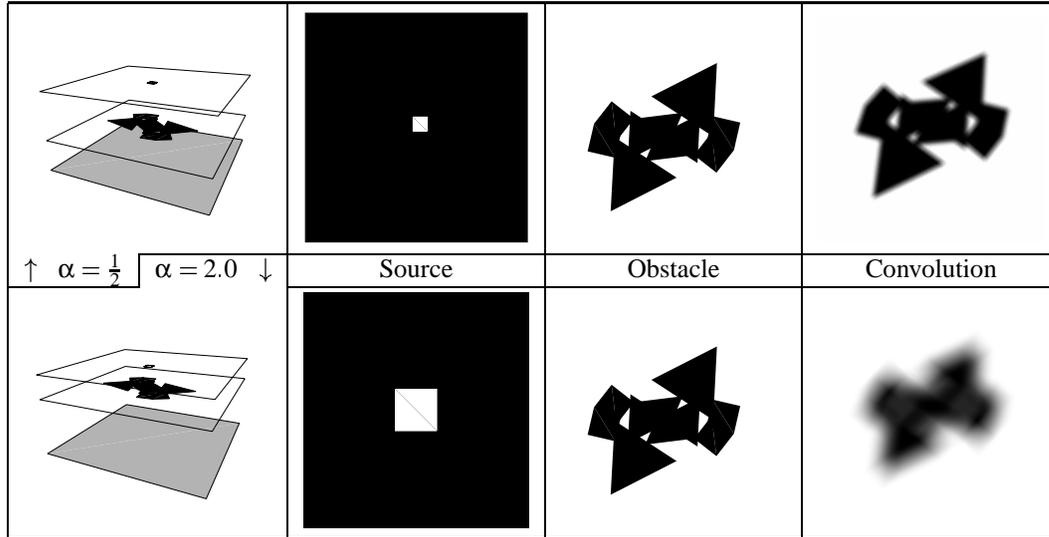


FIG. 7.4: Exemples d'application de la formule de convolution (7.2) pour des configurations correspondant respectivement à $\alpha = \frac{1}{2}$ et $\alpha = 2.0$. Le rapport d'échelles entre les images de la source et de l'obstacle prend en compte les distances relatives entre les trois éléments.

7.2.2 Cas général

Dans le cas plus général d'éléments plans mais non parallèles, il n'est plus possible d'exprimer le terme de visibilité $V(y)$ comme une convolution car la projection de l'obstacle sur la source depuis un point y du récepteur change de forme lorsque y parcourt le récepteur. Si de plus la source, l'obstacle, et le récepteur ne sont pas plans, des phénomènes d'auto-occlusion accentuent encore ces déformations. Nous proposons néanmoins de calculer la texture de visibilité en appliquant la méthode décrite précédemment à un ensemble d'éléments plans et parallèles, que nous appellerons *géométrie virtuelle*, et de transformer la texture obtenue pour représenter le plus correctement possible les ombres sur le récepteur réel. Cela signifie plus précisément que nous devons :

1. Choisir une source, un obstacle et un récepteur virtuels, plans et parallèles. On décide pour cela d'une direction normale D commune aux trois plans portant les éléments virtuels, puis de la position de ces trois plans le long de D . Ils faut ensuite choisir dans ces plans les éléments virtuels qui produiront l'ombre la plus fidèle possible ;
2. appliquer la méthode de convolution à ces éléments virtuels ;
3. transformer et moduler la texture de visibilité pour simuler de façon appropriée l'ombre produite sur le récepteur.

Bien que les éléments virtuels soient choisis de manière à minimiser l'erreur d'approximation qu'ils induisent, cette erreur n'est jamais nulle. Nous verrons à la section 7.5 comment contrôler cette erreur, au prix du calcul de convolutions supplémentaires. Le contrôle de l'erreur est en effet indispensable pour satisfaire les critères de transfert temps de calcul/précision que nous nous sommes fixés.

(a) Construction de la géométrie virtuelle

Comme nous l'avons dit, la direction D portant la normale commune aux plans des objets virtuels est la première chose à déterminer. Ce choix influe de façon évidente sur la nature des approximations réalisées ; nous en discuterons à la section 7.5.1. On peut considérer qu'un choix raisonnable pour D correspond à une direction représentative de la

direction moyenne des droites coupant la source, l'obstacle et le récepteur, comme celle montrée par la figure 7.5.a. Nous discuterons plus précisément du calcul de cette direction dans la partie traitant de l'implémentation.

Une fois la direction D choisie, nous définissons un repère local R' dont l'axe Oz est porté par D , et dont les axes Ox et Oy sont choisis arbitrairement de façon à former un repère orthonormé direct. Le choix des plans portant les éléments virtuels consiste alors à déterminer trois ordonnées z_s , z_b et z_r dans ce repère, comme indiqué sur la figure 7.5.b. L'influence de ces valeurs sur l'erreur d'approximation sera également discuté de façon précise à la section 7.5.1.

Les plans virtuels étant choisis, les éléments virtuels sont obtenus par projection des éléments réels sur ces plans (Figure 7.5.c) ; les projections sont choisies de façon à respecter au mieux la géométrie des objets réels :

- la source virtuelle est la projection orthogonale de la source réelle sur P_s . La source virtuelle conserve donc globalement l'aspect de la source réelle, vue du récepteur.
- l'obstacle virtuel est l'image sur le plan de l'obstacle virtuel de l'obstacle réel, par une projection centrale depuis un point de la source. Vu de ce point là, l'obstacle virtuel et l'obstacle réel ont donc exactement le même aspect.
- Le récepteur virtuel est obtenu de la même façon que l'obstacle virtuel. Bien que sa forme n'ait pas d'influence directe sur le calcul de la convolution, nous verrons qu'elle est prise en compte lors de l'optimisation du nombre de polygones effectivement utilisés pour former l'obstacle virtuel.

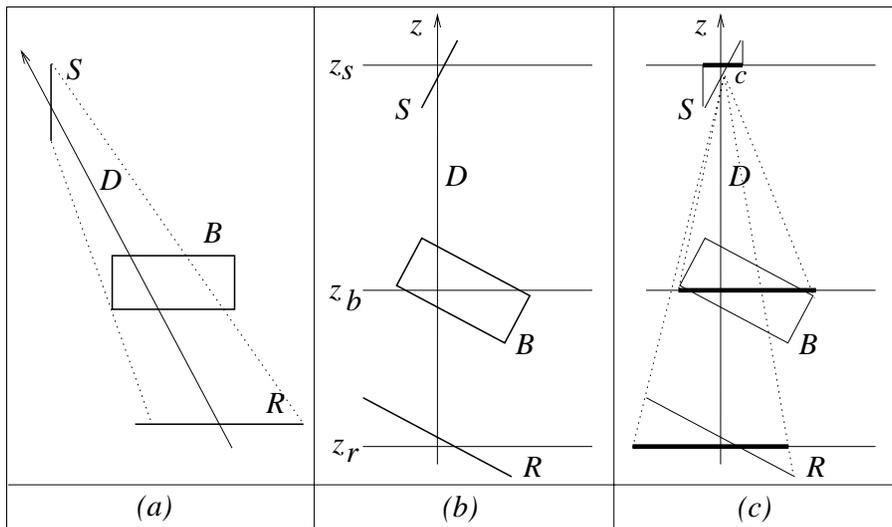


FIG. 7.5: Construction de la géométrie virtuelle : (a) Détermination de la direction D portant la normale aux plans virtuels. (b) Choix des ordonnées z_s , z_b et z_r définissant la position des plans virtuels. (c) projection des éléments réels sur ces plans pour obtenir leurs homologues virtuels.

(b) Utilisation de la texture de visibilité sur le modèle réel

Disposant des éléments virtuels plans et parallèles, la méthode de convolution peut être appliquée pour calculer l'ombre correspondante. La texture de visibilité obtenue est ensuite projetée sur le récepteur réel depuis le point c , où elle est modulée par l'éclairage direct $\rho EF_S(y)$ que produit la source réelle sur le récepteur réel.

La figure 7.6 montre le processus entier appliqué à une scène réelle.

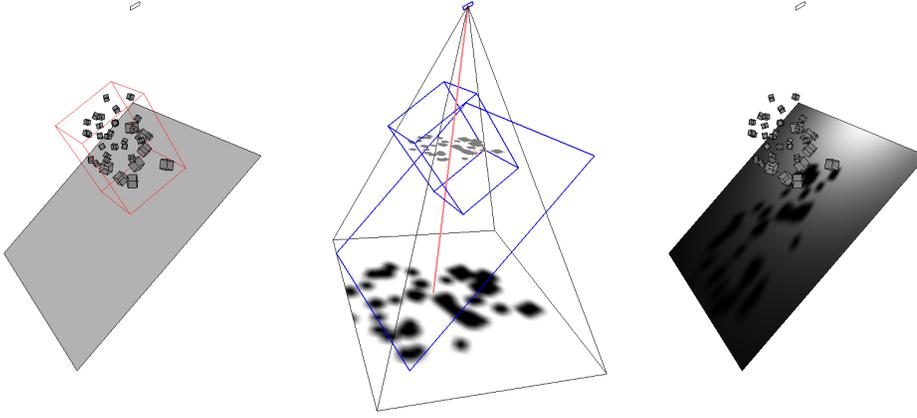


FIG. 7.6: Utilisation de la géométrie virtuelle pour une scène 3D. A gauche : les objets réels. Au centre : l'obstacle et le récepteur virtuel, plus la texture de visibilité. A droite : la texture de visibilité projetée et modulée par l'éclairage direct sur le récepteur réel.

7.3 Extensions directes

Nous présentons dans cette section un certain nombre de configurations dans lesquelles le calcul des ombres par convolution s'applique au prix de menues modifications de l'algorithme présenté à la section précédente. Les différentes situations peuvent être combinées pour traiter plusieurs de ces situations en même temps.

Les extensions sont toujours présentées dans le cas d'école introduit en 7.2.1 pour plus de simplicité. À l'exception de la dernière (section 7.3.6), elles se généralisent aisément à des éléments non plans ou non parallèles.

7.3.1 Sources non uniformes

Lorsque sur la source n'est pas uniforme, la méthode que nous avons présentée s'applique toujours. Si l'on désigne par $e(x)$ l'émittance sur la source au point x , on a alors en tout point y du récepteur :

$$b(y) = \frac{\rho}{\pi} \int_S \frac{\cos \theta \cos \theta'}{r^2} e(x) v(x, y) dx$$

Le calcul conduit page 102 pour l'obtention de l'équation (7.2) reste valable si l'on remplace tout du long S par $S \times E$. Le terme $V(y)$ devient alors :

$$\begin{aligned} V(y) &= \int_{\mathbb{R}^2} S(x) E(x) v(x, y) dx \\ &= \frac{1}{\alpha^2} (s'_\alpha * p_{1+\alpha}) \end{aligned}$$

En posant :

$$s'_\alpha(x) = S \left(-\frac{1}{\alpha} x \right) e \left(-\frac{1}{\alpha} x \right)$$

Le terme de visibilité s'écrit donc toujours comme une convolution mais ne représente plus à proprement parler de la visibilité.

Le calcul pratique s'effectue en remplaçant l'image binaire de la source par une image dont les pixels varient dans l'intervalle $[0, 1]$ de manière à représenter la fonction e normalisée. L'image obtenue contient donc les valeurs d'échantillonnage de la fonction s'_α divisée

par sa valeur maximale sur toute la source. La constante de normalisation est ensuite incorporée à la composante d'éclairage direct, pour obtenir l'éclairage désiré. La figure 7.7 montre un exemple d'application de notre méthode à une source non uniforme faite de quatre sources plus petites.

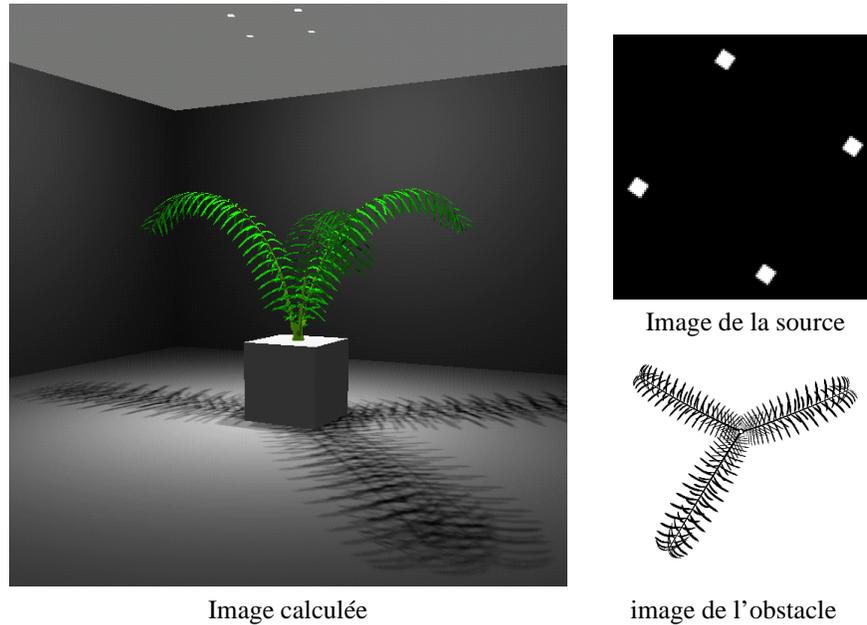


FIG. 7.7: Exemple d'extension de la méthode de convolution à une source non uniforme. L'ensemble formé des quatre petites sources a été considéré comme une unique source non uniforme, produisant automatiquement la superposition des quatre ombres du palmier.

7.3.2 Sources volumiques

Lorsque la source n'est pas un polygone mais un objet lumineux plus complexe, comme par exemple un luminaire modélisé par un ensemble de polygones, nous calculons l'image de la source de la même façon que pour un polygone, sachant que nous commettons une erreur d'approximation supplémentaire, qui sera discutée à la section 7.5.2.(c).

La figure 7.8 montre un exemple d'image calculée pour une source non polygonale, formée d'un assemblage de quatre tubes néon, chacun constitué de 360 polygones. On observe clairement la variation par paliers de l'illumination dans la direction perpendiculaire à l'axe des néons et la variation plus régulière dans l'autre direction.

7.3.3 Récepteur non polygonal

Nous nous intéressons maintenant au cas où le récepteur est formé, non plus d'un unique polygone mais d'un ensemble de polygones. Ce cas se présente chaque fois que le récepteur est un objet complexe modélisé par une liste de facettes.

Une unique texture de visibilité est alors calculée pour tout le récepteur et se trouve partagée par chacun de ses polygones. Seul l'éclairage direct est calculé séparément pour chaque polygone du récepteur, puisqu'il faut tenir compte de l'orientation par rapport à la source de chacune des facettes. La figure 7.9 illustre cette extension.

Il faut noter que notre algorithme s'étend naturellement à des surfaces paramétrées, rendues par exemple par lancer de rayon, à condition de calculer les coordonnées de texture pour chaque point afin d'en déduire la couleur exacte.

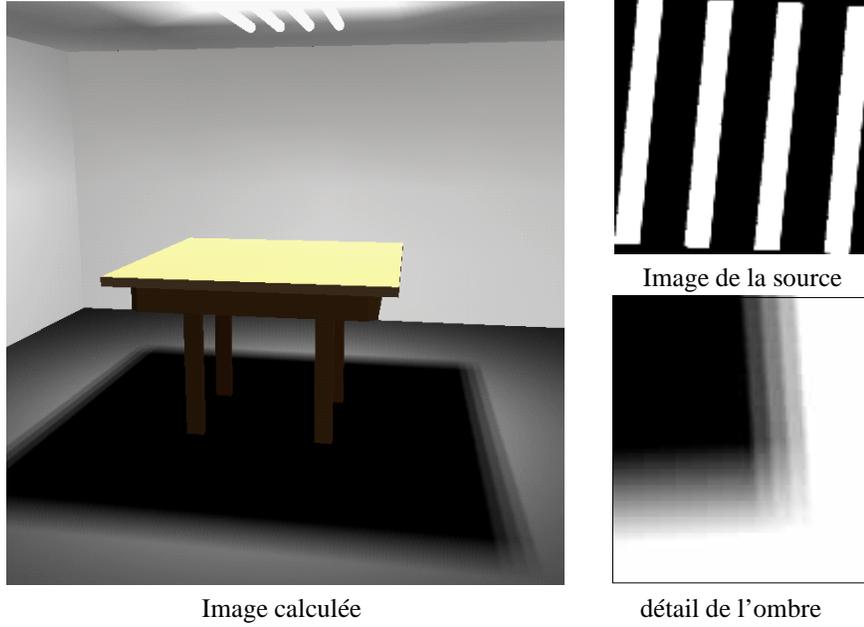


FIG. 7.8: Extension de la méthode de convolution à une source non plane. Dans cet exemple, la source est constituée de quatre tubes néon d'environ trois cent polygones chacun. L'image en bas à droite montre un détail de la texture de visibilité correspondant à l'ombre du coin de la table.

En revanche, comme on peut le voir sur la figure 7.9, la méthode ne simule pas automatiquement l'ombre que porte le récepteur sur lui-même. Nous verrons au chapitre 8 (section 8.5.2) diverses méthodes pouvant résoudre ce problème.

7.3.4 Source à l'infini

Une *source à l'infini* est une source située suffisamment loin et de taille suffisante pour que l'on puisse considérer qu'elle occupe toujours le même ensemble de directions incidentes, vue de n'importe quel point du récepteur. Une telle source peut donc être représentée par un ensemble de directions Δ mesuré par un angle solide Ω (L'exemple couramment cité de source à l'infini est le soleil).

Nous allons montrer que notre méthode s'applique naturellement au cas d'une source à l'infini. L'utilisation d'une convolution pour calculer les ombres dans cette situation particulière a été introduite par Max en 1991 [Max91].

En utilisant l'équation 1.1, nous pouvons écrire que :

$$b(y) = \frac{E\rho}{\pi} \int_{\Delta} \cos\theta v(\omega, y) d\omega$$

En renouvelant l'approximation de la corrélation entre le noyau de radiosité et le terme de visibilité, nous obtenons alors :

$$F_S(y) = E \int_{\Delta} \cos\theta d\omega \quad \text{et} \quad V(y) = \int_{\Delta} v(\omega, y) d\omega$$

Il ne nous reste plus qu'à définir une fonction S représentant la source. Pour tout point x dans le plan d'altitude 1 posons (C.f figure 7.10) :

$$S(x) = \begin{cases} 1 & \text{si } (y, x) \in \Delta \\ 0 & \text{sinon.} \end{cases}$$

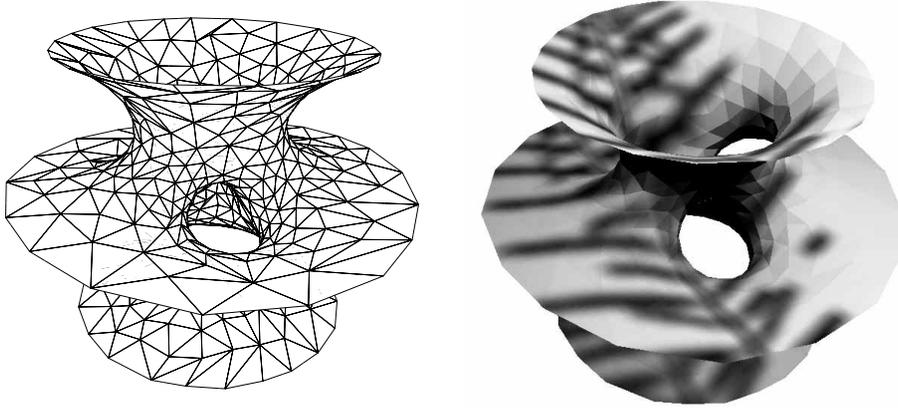


FIG. 7.9: Exemple de récepteur non polygonal. La surface (à gauche) a été placée sous la plante de la figure 7.7. Une unique texture de visibilité est partagée (à droite) par tous les points de la surface (Une seule des quatre sources a été utilisée pour plus de clarté).

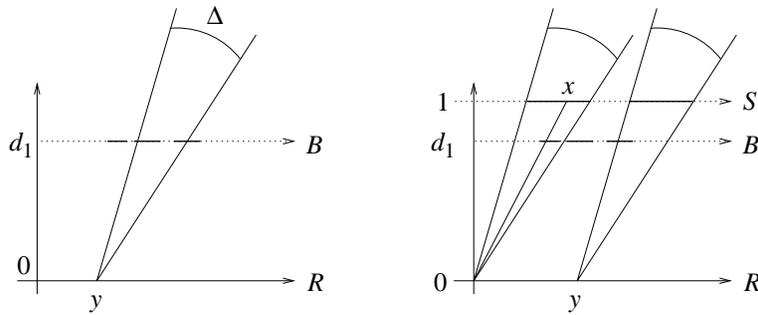


FIG. 7.10: A gauche : Une source à l'infini est représentée par un ensemble de directions Δ d'angle solide Ω . A droite : définition d'une fonction S dans le plan d'altitude 1, pour représenter la source.

Posons maintenant $s_{d_1}(x) = S(-\frac{1}{d_1}x)$. Nous avons alors :

$$\begin{aligned}
 V(y) &= \int_{\mathbb{R}^2} S\left(\frac{1}{d_1}x\right)P(x+y)dx \\
 &= \int_{\mathbb{R}^2} s_{d_1}(-x)P(x+y)dx \\
 &= \int_{\mathbb{R}^2} s_{d_1}(x)P(y-x)dx \\
 &= (s_{d_1} \star P)(y)
 \end{aligned}$$

Il s'agit toujours d'une convolution entre une image de la source et une image de l'obstacle. Seule l'image de la source doit être mise à l'échelle pour tenir compte de la distance entre l'obstacle et le récepteur.

7.3.5 Obstacle semi-transparent

Lorsque l'obstacle est semi transparent mais ne modifie pas la trajectoire des rayons lumineux, son ombre peut encore être approchée en utilisant une convolution. Il suffit pour

cela de remplacer la fonction P à valeur dans $\{0, 1\}$ par une fonction p à valeur dans $[0, 1]$, et qui représente la *transmittance* de l'obstacle.

Cette extension ne s'applique pas lorsque l'obstacle dévie de façon non linéaire les rayons lumineux. C'est le cas par exemple d'une lentille opérant par réfraction.

7.3.6 Éclairage direct

Lors de l'établissement de la formule de convolution (7.2) page 102, le fait d'avoir décorrélé le terme de visibilité du noyau de la radiosité nous a permis d'exprimer le terme de visibilité comme une convolution. Nous allons voir qu'il est possible d'en faire de même avec le terme d'éclairage direct restant, toujours dans le cas particulier où la source et le récepteur sont plans et parallèles.

Notons d la distance entre la source et le récepteur. En utilisant les notations de la figure 7.11.a, la radiosité en un point y du récepteur s'écrit :

$$B(y) = \rho E \int_S \frac{\cos \theta \cos \theta'}{\pi r^2} dx$$

Indépendamment de y , nous savons de plus que $\theta = \theta'$, car la source est parallèle au récepteur et pour la même raison θ vérifie $\cos \theta = \frac{d}{r}$. Nous avons :

$$\begin{aligned} B(y) &= \frac{\rho E}{\pi} \int_S \frac{d^2}{r^4} dx \\ &= \frac{\rho E}{\pi} \int_{\mathbb{R}^2} g_S(x) \frac{d^2}{r(x,y)^4} \\ &= \frac{\rho E}{\pi} \int_{\mathbb{R}^2} g_S(x) \frac{d^2}{(d^2 + (x-y)^2)^2} \\ &= \frac{\rho E}{\pi d^2} (g_S * K_d)(y) \end{aligned}$$

Il s'agit donc d'une convolution, dont le noyau K_d vaut :

$$K_d(x) = \frac{1}{\left(1 + \frac{\|x\|^2}{d^2}\right)^2}$$

La figure 7.11.b donne une représentation graphique de K_d .

Le lecteur pourra vérifier facilement que le noyau K_d est de classe C^∞ et qu'il possède la propriété suivante :

Pour tout réel α l'ensemble des points x tels que $K_d(x) \geq \alpha$ est un compact convexe de \mathbb{R}^2 .

Cette propriété nous a permis de démontrer la conjecture d'*unimodalité* [Hol96, Dre94] du facteur de forme dans le cas d'une source plane, convexe et parallèle au récepteur. Cette conjecture stipule dans le cas général que la radiosité due à une source convexe uniforme n'admet qu'un seul maximum sur un plan d'orientation quelconque. La démonstration de cette propriété dans le cas particulier cité plus haut figure à l'annexe B du présent document.

7.4 Mise en œuvre

Dans cette section nous traitons diverses questions relatives à la mise en œuvre pratique de la méthode de convolution.

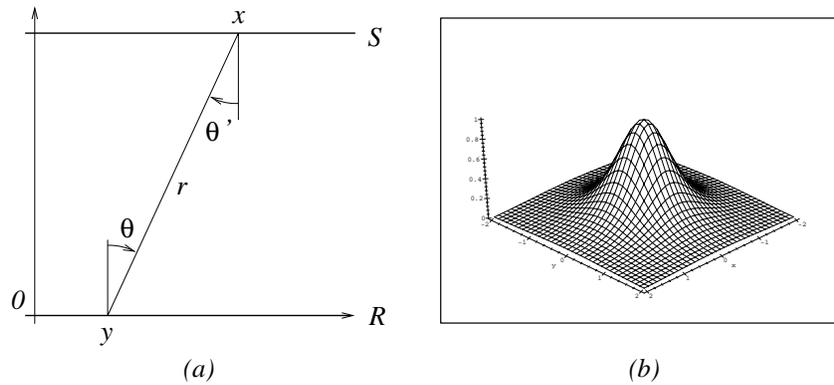


FIG. 7.11: (a) Notations pour le calcul de l'éclairage direct : S désigne le plan de la source, R celui du récepteur et d leur distance mutuelle. (b) Noyau de convolution intervenant dans le calcul de l'éclairage direct.

Nous avons implémenté cette technique dans une plate-forme d'expérimentation d'algorithmes de radiosité hiérarchique (*BRIGHT*), ce qui nous a permis d'une part de bénéficier de la présence d'algorithmes de calcul de l'éclairage direct, et d'autre part de disposer des obstacles (également des sources et récepteurs, dans le cadre des extensions énumérées plus haut) sous la forme d'une hiérarchie de boîtes englobantes (où *clusters*).

Nous avons également implémenté un algorithme de sélection automatique des configurations dans lesquelles la méthode de convolution peut être avantageusement appliquée. Nous reparlerons au chapitre 8 de cette technique dans le cadre de l'utilisation automatique de la méthode de convolution pour accélérer les méthodes hiérarchiques de radiosité.

Nous supposons donc disposer d'une source, d'un récepteur et d'un obstacle sous la forme d'un *cluster* et nous allons discuter des différents problèmes liés au choix de la géométrie virtuelle, au calcul pratique des images destinées à la convolution ainsi qu'au calcul de la convolution proprement dit, avant de terminer cette section en expliquant la modulation et le rendu des textures de visibilité calculées.

7.4.1 Du choix de la géométrie virtuelle

Le choix de la géométrie virtuelle correspond à la détermination de la direction D puis au choix des ordonnées z_s, z_b et z_r des plans des objets virtuels, le long de D .

(a) Choix de la normale aux plans virtuels

Comme nous l'avons vu plus haut, la direction D doit représenter la direction moyenne des droites coupant la source et le récepteur. L'emploi d'un terme aussi peu précis est justifié par le fait que le choix de la direction D n'affecte pas l'emplacement des ombres, mais seulement leur gradient. Nous verrons cependant que dans certains cas, comme lors de la subdivision du récepteur, il peut être important de choisir D selon des critères plus restrictifs. La méthode que nous employons se borne donc à choisir une direction dans un ensemble le plus représentatif possible des droites coupant la source et le récepteur.

Lorsque la source est un polygone, nous déterminons tout d'abord un *récepteur utile* (C.f figure 7.12) en ne considérant du récepteur que la partie qui voit un seul coté de la source. La partie restante du récepteur ne recevant pas de lumière. Nous effectuons le même tri pour la source lorsque le récepteur est un polygone : la *source utile* est définie comme l'ensemble des points de la source qui voient la face réceptrice du récepteur.

La direction D est ensuite déterminée comme joignant les centres de gravité de la source utile et du récepteur utile. Cette façon de procéder évite que D soit parallèle à la

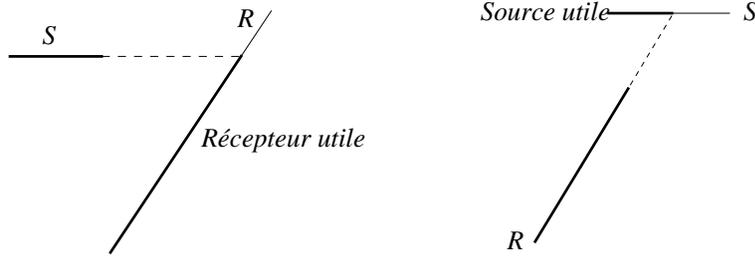


FIG. 7.12: Schéma de principe de la détermination de la source utile et du récepteur utile.

source, ce qui compromettrait l'échantillonnage de la source, ou parallèle au récepteur, ce qui compromettrait la projection de la texture de visibilité sur celui-ci.

(b) Choix des plans virtuels

Les altitudes respectives des plans des objets virtuels affectent directement la dureté des ombres simulées, puisque le coefficient α s'exprime comme

$$\alpha = \frac{z_b - z_r}{z_s - z_b}$$

Choisir les altitudes z_s, z_b et z_r revient donc à figer la dureté des ombres simulées par notre méthode. Nous verrons à la section 7.5.4 qu'il est possible d'exprimer la taille caractéristique des zones de pénombre dans la texture de visibilité, ainsi que ses valeurs minimales et maximales en fonction des valeurs $z_b^{\min}, z_b^{\max}, z_r^{\min}, z_r^{\max}, z_s^{\min}$ et z_s^{\max} qui sont les altitudes minimales et maximales de l'obstacle, du récepteur et de la source le long de D . Nous commençons donc par choisir z_s et z_r comme :

$$z_s = \frac{1}{2} (z_s^{\min} + z_s^{\max}) \quad z_r = \frac{1}{2} (z_r^{\min} + z_r^{\max})$$

Le choix le plus judicieux pour z_b apparaît alors être celui pour lequel la taille caractéristique des zones de pénombre produites est égale à la moyenne des tailles extrémales normalement présentes dans l'ombre exacte. Nous avons montré que la valeur correspondante de z_b doit vérifier :

$$z_b = \frac{D_1 z_b^{\min} + D_2 z_b^{\max}}{D_1 + D_2} \quad \text{avec} \quad D_1 = z_s - z_b^{\max} \quad \text{et} \quad D_2 = z_s - z_b^{\min}$$

7.4.2 Calcul des images pour la convolution

N'ayant besoin que d'images de la source virtuelle et de l'obstacle virtuel, il n'est en fait pas nécessaire de calculer explicitement la projection des polygones concernés sur les plans virtuels avant de former des images représentant leur fonction caractéristiques s_α et $p_{1+\alpha}$. Ces images peuvent en effet être obtenues directement à partir des objets réels et de façon très efficace, en utilisant les capacités graphiques des ordinateurs : l'image de l'obstacle virtuel n'est autre que ce qui apparaît à l'écran lorsque l'on affiche l'obstacle réel vu du point c (voir figure 7.13). Il en est de même pour l'image de la source virtuelle, mais en utilisant une perspective orthographique. Le rendu de ces images est d'autre part effectué en mémoire (rendu *off-screen*) plutôt qu'à l'écran.

Pour obtenir des images représentant de façon appropriée l'échantillonnage des fonctions s_α et $p_{1+\alpha}$ il faut cependant utiliser les réglages suivants :

- La source est rendue en blanc sur fond noir, l'obstacle est rendu en noir sur fond blanc ;

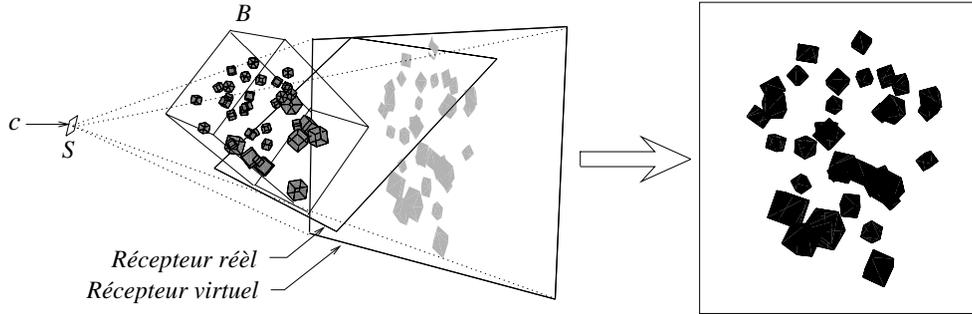


FIG. 7.13: Échantillonnage de $p_{1+\alpha}$ par rendu off-screen de l'obstacle réel

- le réglage de la taille des volumes de projection de la source et de l'obstacle permettent d'obtenir directement le rapport d'échelle de $\frac{\alpha}{1+\alpha}$ entre les deux images ;
- la projection orthographique de la source doit être composée avec une symétrie autour de l'axe de projection, afin de tenir compte du signe négatif apparaissant dans la définition de s_α . Cette opération revient à renverser l'image obtenue autour de son centre ;
- aucune gestion des parties cachées (*z-buffer*) n'est nécessaire.

Nous allons maintenant discuter de façon plus précise trois questions essentielles relatives au calcul de ces images : Le choix de leur résolution d'échantillonnage, la sélection du volume de projection servant à calculer l'image de l'obstacle virtuel et l'*anti-aliasage* des images obtenues.

(a) Choix de la résolution

Le choix de la résolution d'échantillonnage des images de la source et de l'obstacle virtuels est crucial car cette résolution détermine la taille de la texture de visibilité que l'on calcule et donc le coût de l'algorithme (*C.f* section 7.6). Les deux images ne sont pas nécessairement de la même taille, mais étant donné le rapport d'échelle $\frac{\alpha}{1+\alpha}$ qui relie les deux images, l'échantillonnage de la source et de l'obstacle sont proportionnels. Nous choisissons donc la taille de l'image de l'obstacle virtuel la plus petite possible sachant que :

1. La source et l'obstacle doivent être représentés avec suffisamment de pixels pour éviter de trop gros problèmes d'aliasage (Voir la section (c)) ;
2. l'échantillonnage doit suffire à représenter la variation de visibilité au niveau du récepteur. Comme nous le verrons à la section 7.5.2, le gradient g de la visibilité sur le récepteur est en première approximation inversement proportionnel à α et à la taille de la source. Connaissant $\|g\|$ (en unités d'intensité lumineuse par unité de longueur), la résolution $N \times N$ de la texture est suffisante si

$$\|g\| < \frac{I_{max} N}{2^p L}$$

Dans cette expression, L représente la taille de la texture de visibilité en unités de longueur (L unité choisie importe peu, puisqu'elle disparaît dans le rapport), p est le nombre de bits servant à coder les niveaux d'intensité lumineuse (En pratique $p = 8$), et I_{max} l'intensité maximale représentable (I_{max} vaut 1.0) ;

3. la taille de la texture doit être adaptée à celle du récepteur : bien que cela dépende du point de vue utilisé pour l'affichage, le rapport entre la taille des polygones et la taille des textures plaquées dessus doit être à peu près uniforme.

Au vu de ces considérations, les tailles de textures utilisées varient en pratique de 32×32 pour les plus petits polygones (il n'y a en fait pas de limite inférieure, si ce n'est qu'une texture de visibilité soit plus intéressante qu'une interpolation linéaire à partir des valeurs d'illumination aux sommets des polygones) à 512×512 pour les très grands polygones (murs, sols,...). D'une façon générale, il est assez rare qu'une texture de taille 256×256 soit insuffisante.

(b) Sélection du volume de projection de l'obstacle

Le cône de projection (ou *Frustum*) de l'obstacle est l'ensemble des points de l'espace susceptibles de contribuer à l'image de l'obstacle. La projection utilisée étant une projection centrale, cet ensemble est un cône tronqué par deux plans (C.f figure 7.14.a). La direction normale à ces plans n'est autre que la direction D intervenant dans la définition des éléments virtuels.

Pour effectuer la projection *off-screen* de l'obstacle, on utilise un volume de projection le plus petit possible afin d'une part de limiter le nombre de polygones à rendre et d'autre part d'obtenir la résolution la plus fine possible dans la texture.

On définit pour cela le *cône de projection utile* de l'obstacle comme étant l'intersection entre le cône de projection (non tronqué) du récepteur et celui (tronqué) de l'obstacle. Ces cônes sont calculés par projection des sommets des boîtes englobantes de ces objets sur le plan P_{far} , et leur intersection est calculée par intersection des intervalles $[x_{min}, x_{max}]$ et $[y_{min}, y_{max}]$ correspondants (C.f figure 7.14.b). La conséquence de cette optimisation est que la texture de visibilité calculée ne recouvre pas entièrement le récepteur.

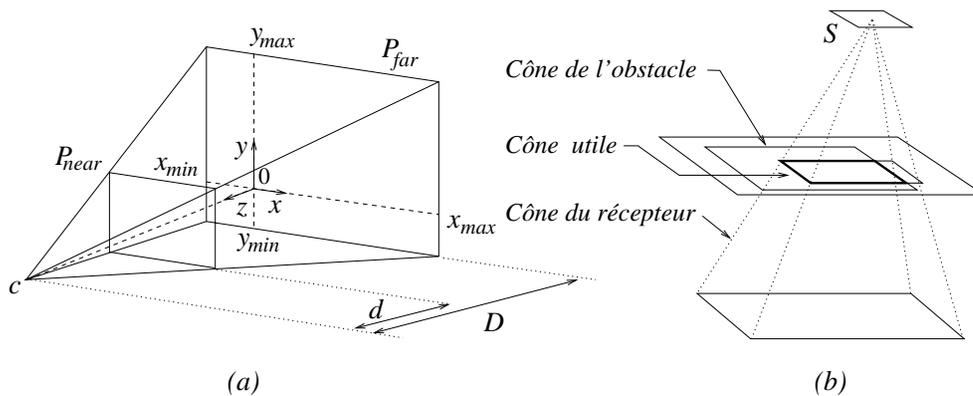


FIG. 7.14: (a) Définition du cône de projection d'un obstacle. Les valeurs $x_{min}, x_{max}, y_{min}$ et y_{max} sont obtenues par projection de la boîte englobante de l'obstacle. (b) Calcul du cône de projection utile de l'obstacle par intersection avec celui du récepteur.

La détermination des plans P_{near} et P_{far} est plus délicate car il se peut que le *cluster* contenant l'obstacle recouvre la source ou le récepteur (sans pour autant les contenir). Comme nous ne voulons prendre en compte que les parties de l'obstacle effectivement susceptibles de contribuer à l'ombre sur le récepteur, nous éliminons d'office, dans le cas où la source ou le récepteur sont plans, tous les polygones de l'obstacle situés du mauvais coté de ces plans. Dans les autres cas, il se peut que $z_{near} \geq D$. La librairie graphique imposant d'autre part que z_{near} soit strictement inférieur à D , nous limitons arbitrairement la position du plan P_{near} à $z_{near} = D - \epsilon$, où ϵ est négligeable devant D .

(c) Anti-aliassage des images

Comme nous l'avons vu, la texture de visibilité calculée ne recouvre pas entièrement le récepteur, mais se restreint à la partie contenant les ombres. La valeur de cette texture

au niveau de sa bordure doit donc correspondre parfaitement à la valeur de visibilité à cet endroit afin d'éviter une éventuelle discontinuité de l'illumination du récepteur sur la frontière de la texture.

D'autre part, la méthode que nous employons pour calculer la texture de visibilité est susceptible de produire une image dont les valeurs sont faussées à cause de l'*aliasage* des images utilisées pour calculer la convolution. En effet, si l'on échantillonne s_α par une image I à valeurs dans $\{0, 1\}$, il y a peu de chances pour que I et s_α aient même valeurs. Le pourcentage d'erreur sur l'image I entière est d'autant plus élevé que la source est représentée par un petit nombre de pixels dans I .

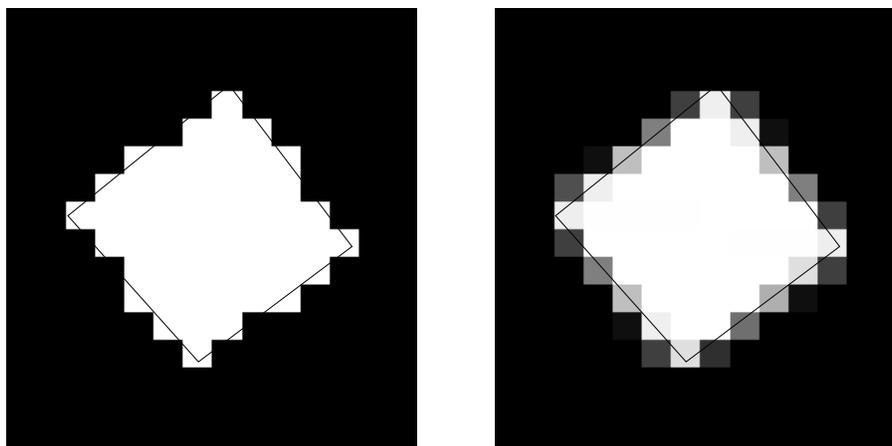


FIG. 7.15: Image de la source correspondant à l'exemple précédent, calculée sans anti-aliasage (à gauche) et avec anti-aliasage (à droite). Le quadrilatère en trait fin représente approximativement la position réelle de la source.

Pour corriger ce problème, nous avons recours à une technique d'*anti-aliasage*, qui consiste à donner à chaque pixel la valeur correspondant à la proportion du pixel effectivement recouverte par un polygone. Cette technique est disponible dans les bibliothèques graphiques comme **OpenGL** et peut être appliquée de façon transparente au moment du rendu des polygones. La figure 7.15 donne un exemple d'application de cette technique sur l'image de la source.

Malheureusement l'anti-aliasage proposé par **OpenGL** ne calcule qu'une approximation de la proportion de chaque pixel couvert par le polygone, selon les implémentations et les machines utilisées [NDW93]. Nous donnons sur la figure 7.16 les pourcentages d'erreurs obtenus après anti-aliasage sur deux machines différentes : O_2 et $Onyx^2$ (C.f figure 7.16). Pour mesurer l'erreur due à l'aliasage, nous avons mesuré la valeur maximale de la texture de visibilité obtenue dans une configuration pour laquelle la textures contient des points qui voient entièrement la source. Cette valeur maximale doit donc être égale à 1. Le tableau sur la figure 7.16 représente le rapport entre la valeur maximale de la texture calculée et cette valeur de référence.

Au vu des mesures effectuées on note tout d'abord que l'erreur est globalement bien réduite par l'anti-aliasage, plus encore sur l' $Onyx^2$ que sur l' O_2 . Comme nous pouvions nous y attendre, l'erreur augmente lorsque α diminue et lorsque N diminue, puisque l'image de la source occupe alors de moins en moins de pixels. La valeur ∞ signifie que l'image de la source est noire car la source est trop petite. Dans ce cas, le fait de rajouter à la main un pixel dont la valeur est l'aire apparente de la source ramène l'erreur à 0%. En revanche nous remarquons la faiblesse de l'algorithme implémenté sur l' $Onyx^2$ pour calculer la proportion du pixel effectivement couverte (l'erreur est de 56%). Nous conseillons donc, lorsque la source occupe moins d'un pixel dans l'image, de mettre à jours la valeur de ce pixel explicitement.

α	0.9			0.5			0.2		
Anti-aliasage	Sans	O_2	$Onyx^2$	Sans	O_2	$Onyx^2$	Sans	O_2	$Onyx^2$
$N = 128$	15%	12%	6%	41%	14%	6%	∞	1%	56%
$N = 256$	14%	7%	3%	21%	7%	3%	20%	8%	3%
$N = 512$	5%	0.7%	1%	4%	3%	0.2%	8%	7%	2%

FIG. 7.16: Efficacité de l'anti-aliasage sur la réduction de l'erreur due à l'échantillonnage de s_α en fonction de la taille des textures (i.e de la précision de l'échantillonnage) et du type de machine utilisée. Au delà de 1%, l'erreur devient visible à l'écran au point de recollement de la texture avec les polygones adjacents.

Finalement l'anti-aliasage est également utilisé pour calculer l'image de l'obstacle virtuel afin de contrer une autre source d'erreur : lorsque l'obstacle contient des parties très fines (Comme les branches du mobile de la figure 7.32) il évite que la pixélisation des détails correspondants n'affecte la texture de visibilité. La figure 7.17 montre l'effet de l'anti-aliasage de l'obstacle sur l'exemple du mobile.

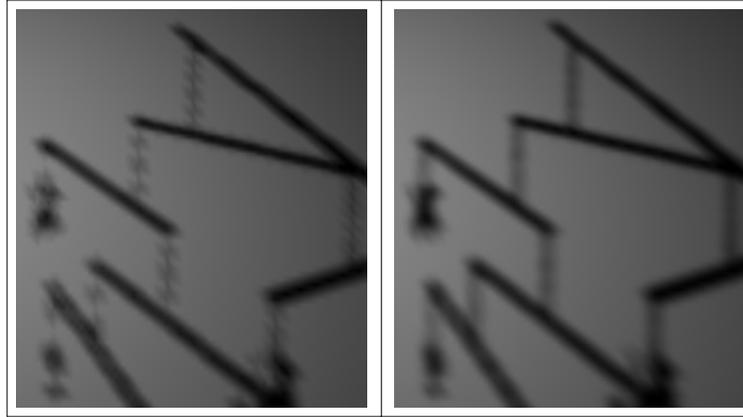


FIG. 7.17: Effet de l'anti-aliasage de l'image de l'obstacle virtuel. A droite : l'anti-aliasage corrige l'effet de pixélisation dû au sous-échantillonnage de l'image de l'obstacle. (A gauche), lorsque celui-ci contient des structures fines.

En conclusion, l'utilisation de l'anti-aliasage lors de l'échantillonnage des fonctions caractéristiques de la source et de l'obstacle ne corrige pas entièrement l'erreur, puisqu'elle n'augmente pas la résolution de l'échantillonnage, mais la réduit fortement, jusqu'à un point où elle est indiscernable à l'œil nu.

7.4.3 Calcul de la convolution

Nous présentons dans cette section deux méthodes de calcul de la convolution entre l'image de la source et l'image de l'obstacle virtuel.

(a) Utilisation de la transformée de Fourier rapide

Le calcul de la convolution est effectué sur la base de la relation liant les transformées de Fourier de deux fonctions et leur convolution. Nous notons F l'opérateur de transformation de Fourier, défini par :

$$F : L^2(\mathbb{C}^2) \rightarrow L^2(\mathbb{C}^2)$$

$$F(f)(y) = \int_{\mathbb{R}^2} e^{-2i\pi(x_1y_1 + x_2y_2)} f(x_1, x_2) dx_1 dx_2$$

Nous avons alors :

$$(f \star g)(y) = F^{-1}(F(f) \times F(g))$$

Le calcul des transformées de Fourier des images est effectué par *Transformée de Fourier Rapide (FFT)* normalisée. Bien que ce ne soit pas nécessaire, il est plus pratique que les deux images aient la même taille. Nous effectuons alors par convolution, l'équivalent de trois *FFT* et un produit d'images point à point. Le tableau suivant donne les temps de calculs associés à la transformée de Fourier rapide sur deux machines différentes :

N	128	256	512	1024
O_{nyx^2}	6	25	110	500
O_2	20	50	170	770

FIG. 7.18: Temps en millisecondes pour une *FFT* sur une image de taille $N \times N$ variable, sur O_2 et O_{nyx^2} .

(b) Utilisation de la carte graphique

Certaines machines, notamment les stations O_2 sont capable d'effectuer rapidement des convolutions d'images par un masque de petite taille (De 7×7 à 11×11 , selon les modèles). Il est donc possible de calculer la convolution de deux images en découpant l'une des images en sous-images de taille réduite, servant de filtre pour convolution avec l'autre image. On ajoute ensuite, en les disposant de façon appropriée, les images obtenues pour reconstruire la convolution cherchée.

	$N = 64$		$N = 128$		$N = 256$		$N = 512$	
	O_2	O_{nyx^2}	O_2	O_{nyx^2}	O_2	O_{nyx^2}	O_2	O_{nyx^2}
n=5	3.5	0.012	3.8	0.13	5.2	0.15	11.2	0.15
n=7	3.5	0.014	3.9	0.22	5.3	0.19	11.2	0.31
n=9	63.0	0.012	272.0	0.12	1130.0	0.27	4521.0	0.31
n=11	81.2	0.012	364.1	0.13	1543.0	0.15	6244.0	0.15

FIG. 7.19: Vitesses de calcul des convolutions par l'extension *OpenGL*. Les temps sont mesurés en millisecondes par convolution. La taille de l'image est N et celle du filtre est n .

Les vitesses d'exécution des convolutions pour des filtres et des images de taille variées sur deux machines différentes sont transcrites sur la figure 7.19. Elle inspirent quelques commentaires :

Les temps de calcul sur l' O_{nyx^2} sont relativement uniformes et aucune loi ne s'en dégage clairement. La carte graphique effectue toutes les convolutions à vitesse quasi constante (Les convolutions étant très rapides, on mesure leur vitesse moyenne sur un grand nombre d'opérations, les temps de calcul sont donc de fait fortement perturbés par l'activité du processeur entre ces convolutions.).

En ce qui concerne l' O_2 , l'augmentation brutale des temps de calcul lorsque la taille des filtres dépasse 7×7 témoigne des limites du composant qui leur est dédié : à partir de 9×9 , les convolutions sont effectuées par le processeur principal et leur coût, bien plus élevé par ailleurs, devient clairement proportionnel au nombre de pixels dans le filtre et dans l'image.

En conclusion, cette technique peut être utilisée pour améliorer les performances de notre algorithme dans le cas où α est petit, *i.e* lorsque la source est effectivement représentée dans son image par un petit nombre de pixels, et en fonction de la machine utilisée. On effectue alors autant de convolutions qu'il faut de filtres de la taille choisie, pour recouvrir la source.

(c) Zone de sécurité pour la convolution

Quelle que soit la méthode utilisée (logicielle ou matérielle) l'algorithme de convolution suppose implicitement que les images sont périodiques. Le résultat de la convolution est donc également une image périodique, dans laquelle les pixels correspondant à la superposition d'une image sur la frontière de l'autre (et donc simultanément sur la frontière lui faisant face) n'ont pas de signification en terme de visibilité.

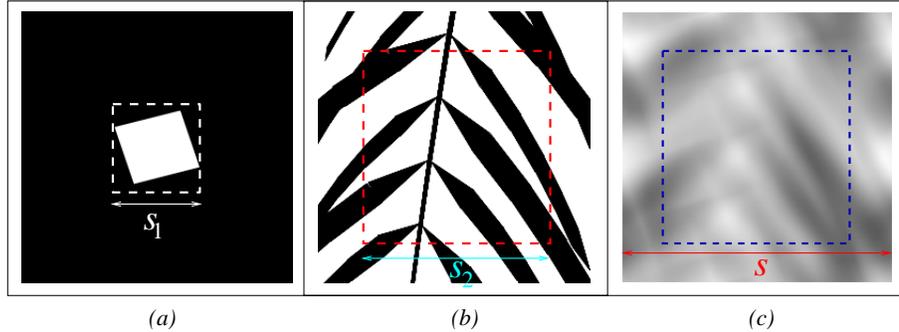


FIG. 7.20: Calcul de la zone des pixels valides dans l'image de convolution. Les volumes de projection de la source et de l'obstacle ont été élargis jusqu'à ce que la taille apparente s_1 de la source dans son image, la taille de la zone utile s_2 de l'image de l'obstacle, et la taille s de l'image de convolution vérifient $s_1 + s_2 \leq s$. Les pixels ayant une signification en termes de visibilité sont ceux à l'intérieur du cadre dans l'image de droite.

Lorsque les images ont même taille, cette remarque signifie que seul le pixel situé au centre de la convolution est valide, puisqu'il correspond à l'unique position dans laquelle les images se superposent sans effet de bord.

En fait, la source n'atteignant jamais les bords de son image, la région des pixels valide est plus étendue. Si l'on note s_1 la taille apparente de la source dans son image (C.f figure 7.20), tous les pixels de l'image de convolution situés à une distance supérieure à s_1 du bord le plus proche sont valides. Connaissant la zone utile dans l'image de l'obstacle (Cette zone est l'intersection du volume de projection utile de l'obstacle avec son plan de projection) et la taille apparente s_1 de la source, les cônes de projection servant à fabriquer les images sont élargis jusqu'à ce que les taille apparentes s_1 et s_2 de la source et de la zone utile de l'image de l'obstacle, et la taille s de l'image de convolution vérifient

$$s_1 + s_2 \leq s$$

En pratique, nous mesurons donc s_1 , s_2 et s une fois tous les autres réglages faits, et nous dilatons les volumes de projection d'un commun facteur $\frac{s}{s_1 + s_2}$.

7.4.4 Rendu des textures

Comme nous l'avons déjà vu, l'image issue de la convolution est utilisée au moment du rendu comme une *texture* sur le récepteur, modulée par l'éclairage direct depuis la source. Nous allons traiter différents aspects de cette opération, en commençant par le calcul des valeurs d'illumination directe.

(a) Modulation des textures

Les valeurs d'éclairage direct sur le récepteur sont calculées analytiquement en des points uniformément distribués sur celui-ci grâce à la formule (1.11) page 26. Ces valeurs sont ensuite stockées en mémoire dans une structure appelée *maillage d'affichage*. Au moment du rendu, un tel maillage peut être affiché de façon très efficace par **OpenGL**

qui n'effectue les calculs de projection pour chaque nœud du maillage qu'une seule fois, en reconstituant les polygones à partir de leurs seuls sommets. La figure 7.21 montre des maillages d'affichage sur un quadrilatère et un triangle.

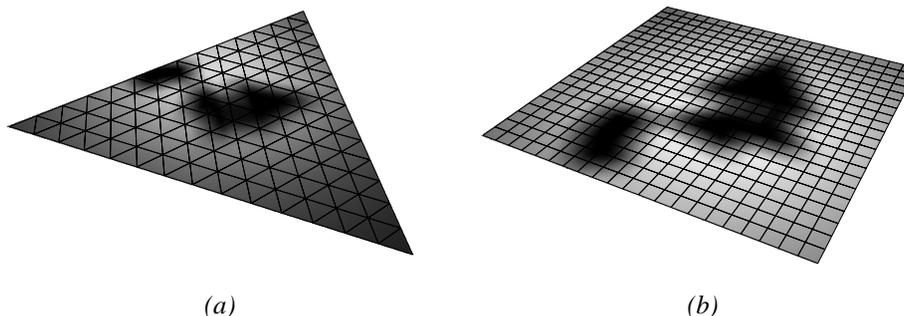


FIG. 7.21: Exemples de maillages d'affichage pour des textures de visibilité (a) sur un triangle et (b) sur un quadrilatère. La résolution du maillage est au moins 10 fois plus faible que celle de la texture.

Pour chaque nœud du maillage il faut également fournir à la librairie graphique des *coordonnées de texture*, qui spécifient l'emplacement du point affiché dans la texture. Lorsque la texture ne recouvre pas entièrement le récepteur, certains points n'ont pas d'image dans la texture. Il est alors possible de préciser à **OpenGL** d'utiliser une couleur spéciale, ou de répéter la valeur du bord de la texture le plus proche. Dans tous les cas, cette valeur doit être 1, puisque le point concerné n'est plus dans l'ombre. Pour un résultat visuel plus satisfaisant, on peut également demander que les valeurs des pixels soient calculées par interpolation (bilinéaire, par exemple) entre les pixels les plus proches dans la texture.

Lorsque **OpenGL** affiche un élément du maillage connaissant les coordonnées de ses sommets ainsi que leurs coordonnées de texture, il calcule les coordonnées des autres points après projection ainsi que leurs coordonnées de texture par interpolation bilinéaire des valeurs aux sommets. Malheureusement, comme nous le verrons à la section suivante, les coordonnées de texture ne varient pas linéairement sur le récepteur, sauf si celui-ci est orthogonal à D . Le maillage d'affichage doit donc être suffisamment fin pour corriger la déformation due à l'usage abusif de l'interpolation bilinéaire. En pratique nous utilisons des maillages d'affichage de taille allant de 2×2 à 20×20 pour les plus grands polygones.

On pourra noter que le calcul de l'éclairage direct, peut être réalisé par d'autres méthodes, par exemple basées sur l'utilisation des sources lumineuses de la librairie graphique **OpenGL**. Alors que les ombres sont visuellement affectées par la substitution abusive de sources ponctuelles à des sources étendues, la composante directe de l'éclairage peut en effet être efficacement simulée par des lampes ponctuelles [HH97].

Nous allons maintenant discuter du calcul des coordonnées de texture des nœuds du maillage d'affichage.

(b) Calcul des coordonnées de texture

Une fois calculée, l'image de convolution doit être projetée sur le récepteur réel et positionnée correctement. Nous calculons pour cela les coordonnées de texture d'un point P du récepteur comme étant les coordonnées de sa projection p sur le récepteur virtuel, depuis le centre de la source, dans le repère (u, v) défini comme suit (C.f figure 7.22) :

Les axes u et v sont parallèles aux bords de l'image de l'obstacle. Le point c_t , porté par la droite reliant les centres de la source et de l'image de l'obstacle virtuel, a pour

coordonnées $(0.5, 0.5)$ sachant que les dimensions réelles de l'image de convolution sur le récepteur virtuel sont $\frac{\alpha}{1+\alpha}$ fois celles de l'image de l'obstacle virtuel.

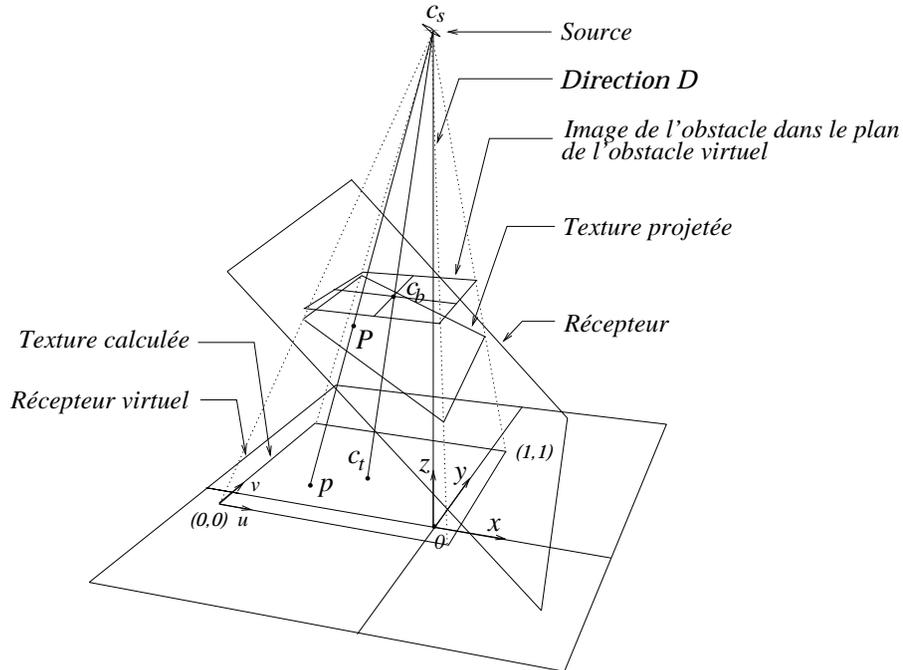


FIG. 7.22: Calcul des coordonnées de texture pour un point P du récepteur. Les points c_s, c_b et c_t sont les centres des images de la source de l'obstacle et de la convolution.

Pour effectuer cette opération, deux solutions s'offrent à nous.

Une première méthode consiste à calculer explicitement les coordonnées de texture des nœuds du maillage d'affichage et à les stocker avec celui-ci. Au moment du rendu, nous fournissons donc directement au moteur graphique les bonnes coordonnées de texture.

La bibliothèque graphique **OpenGL** permet d'autre part d'effectuer le calcul des coordonnées de texture automatiquement à partir des coordonnées 3D de chaque point P . Il faut pour cela lui fournir la matrice de la projection de centre c_s (une matrice 4×4 en coordonnées homogènes) qui n'est autre que celle déjà utilisée pour le calcul de l'image de l'obstacle virtuel. Nous la récupérons en interrogeant *OpenGL* au cours de cette opération. En utilisant cette méthode, nous économisons le stockage des coordonnées de texture.

Nous avons expérimenté les deux méthodes de calcul. Bien que la seconde soit plus efficace, la première n'affecte pas sensiblement les temps de calculs.

(c) Sur-modulation des textures

Les valeurs utilisées par **OpenGL** pour moduler les textures sont des spectres de couleurs de l'intervalle $[0, 1]^3$, une valeur de 0 représentant l'absence totale de sa longueur d'onde de base associée, dans la couleur finale, et 1 sa pleine contribution. Malheureusement, les valeurs d'éclairage direct calculées n'ont aucune raison d'appartenir à l'intervalle $[0, 1]$, notamment parce qu'elles dépendent linéairement de la puissance des sources lumineuses. Elles sont donc automatiquement ramenées à 1 lorsqu'elles sont supérieures à 1, par la bibliothèque graphique elle-même. Il se peut donc en certains points que :

$$V(y) < 1 \quad \rho(y)F_S(y) > 1 \quad \text{et} \quad V(y)\rho(y)F_S(y) \leq 1$$

Dans un tel cas, **OpenGL** affiche un pixel d'intensité inappropriée $V(y)$.

Pour contourner ce problème, nous affichons n fois la texture en ajoutant à chaque fois les valeurs des pixels par *blending*. Les valeurs d'éclairage direct sont, elles, divisées par n . La plus petite valeur de n convenable vaut

$$n = 1 + \lfloor I_{max} \rfloor$$

où $\lfloor x \rfloor$ désigne le plus grand entier strictement inférieur à x , et I_{max} la valeur maximum que prend l'éclairage direct sur les nœuds du maillage d'affichage. En procédant ainsi, tous les pixels sont rendus de façon appropriée.

La méthode que nous avons utilisée a cependant un inconvénient : le fait de diviser par n les valeurs d'éclairage direct divise d'autant la précision avec laquelle elles sont représentées. Une fois l'affichage terminé, les valeurs des pixels ne varient plus dans l'ensemble des valeurs représentables à l'écran (255 niveaux dans chaque couleur de base rouge, vert et bleu) mais dans un ensemble contenant n fois moins d'éléments. Lorsque I_{max} est très grand on observe alors nettement les discontinuités associées au passage d'un niveau à l'autre.

(d) Rendu sur textures

Une texture de visibilité modulée par de l'éclairage direct représente de l'irradiance. Dans de nombreuses applications, l'irradiance est transformée en radiosité par utilisation de textures de réflectance. Dans ce cas la réflectance ρ devient une fonction de la position sur le récepteur.

Pour rendre une valeur uniforme d'irradiance sur un polygone texturé, on module habituellement la texture de réflectance par cette valeur. Dans notre cas la valeur de radiosité n'est pas uniforme et est déjà représentée comme une texture. Nous effectuons donc le rendu en deux passes :

1. la texture de radiosité $y \mapsto V(y)F_S(y)$ est affichée sur le polygone.
2. la texture de réflectance est affichée par dessus de telle sorte que les valeurs des pixels soient multipliées point par point. Nous obtenons alors un affichage de $y \mapsto \rho(y)F_S(y)V(y)$. Nous utilisons pour cela le procédé de *blending* permettant de moduler la valeurs des pixels rendus par celle de ceux déjà présents dans l'image :

```
glEnable(GL_BLEND)
glBlendFunc(GL_DST_COLOR, GL_ZERO)
```

7.5 Contrôle de l'erreur

Bien qu'elle produise des images visuellement satisfaisantes, la méthode de convolution telle que nous l'avons présentée jusqu'ici ne calcule qu'une approximation de la visibilité en tout point du récepteur, notamment parce qu'elle effectue ce calcul à partir d'une représentation simplifiée des éléments (*i.e* la géométrie virtuelle) qui consiste entre autres à remplacer des objets quelconques par des objets plats parallèles.

Après une énumération qualitative de l'erreur d'approximation dans la méthode de convolution, nous donnerons une méthode d'estimation puis de contrôle de cette erreur.

7.5.1 Discussion qualitative des sources d'erreur

De nombreuses approximations interviennent tout au long du calcul des ombres par la méthode de convolution. Nous distinguerons trois catégories :

- l'approximation de la corrélation entre le noyau de la radiosité et la visibilité ;
- les diverses approximations dues à l'échantillonnage. Ces approximations concernent le calcul des images de la source et de l'obstacle virtuels, ainsi que le rendu de la texture de visibilité qui est échantillonnée d'une part spatialement par le maillage d'affichage, et d'autre part au niveau des valeurs de modulation, par la représentation discrète des couleurs en machine ;

- les approximations géométriques, dues à l'utilisation de la géométrie virtuelle pour calculer les ombres.

Ces diverses approximations n'ont pas le même impact sur la qualité des images produites. Alors que la première est négligeable dans la quasi-totalité des configurations (voir section 2.1.3), la seconde produit des artefacts comme l'aliassage dont nous avons parlé à la section 7.4.2.(c). Nous pouvons cependant imputer ce type d'erreur dans une large mesure aux limitations matérielles des machines, et les paramètres de résolution seront toujours fixés de façon à obtenir des images ayant une qualité d'échantillonnage satisfaisante. Enfin, les approximations de type géométrique influent de façon significative sur la qualité des ombres calculées. Nous pouvons séparer ces approximations géométriques en deux catégories : celles provenant de l'utilisation d'un récepteur virtuel plan et celles provenant de l'obstacle virtuel lui-même. Nous allons maintenant détailler qualitativement leurs effets respectifs.

Obstacle virtuel

Pour caractériser l'erreur due à l'utilisation d'un obstacle virtuel plan, considérons le cas particulier d'un récepteur plan parallèle à la source. Les différences de structure entre les ombres produites par l'obstacle réel (volumique) et l'obstacle virtuel (plan) peuvent être comprises en termes d'*événements de visibilité* : lors de la projection de l'obstacle réel sur le plan de l'obstacle virtuel, toutes les discontinuités *triple-arêtes* s'appuyant sur une arête de la source et deux arêtes de l'obstacle se fondent en événements de type *arête-sommet* (Pour une définition et une étude exhaustive des événements de visibilité, le lecteur est invité à consulter [DDP97]). Le gradient de la pénombre est donc localement modifié. Dans le cas d'une source parallèle au récepteur, l'ombre produite par l'obstacle virtuelle prend alors un aspect *uniforme* au sens où la taille apparente des zones de pénombre est la même quelle que soit la composante de l'obstacle qui la crée. Nous verrons plus loin que cet effet est d'autant plus apparent que la source est étendue dans le plan perpendiculaire à D et que l'obstacle réel est étendu dans la direction D .

Récepteur virtuel

Considérons maintenant la configuration "duale" de la précédente, dans laquelle l'obstacle est plan et parallèle à la source, le récepteur étant quelconque. Bien que la projection de la texture de visibilité du récepteur virtuel sur le récepteur réel conserve par construction l'emplacement des ombres, elle conserve également le rapport des tailles entre les zones d'ombre et les zones de pénombre. Le long du récepteur réel, ce rapport dépend fortement de la distance du point considéré au plan de la source. L'ombre calculée peut ainsi contenir des zones de pénombre à l'emplacement de zones d'ombre et réciproquement, selon si le point considéré du récepteur réel est plus proche ou plus éloigné de la source que son projeté sur le plan du récepteur virtuel. Cette erreur est d'autant plus grande que le récepteur est étendu le long de D . Elle n'est d'autre part pas corrigée par la projection du récepteur virtuel sur le récepteur réel.

Les erreurs dues aux approximations géométriques sont de loin les plus significatives, et sont d'autre part inhérentes à la méthode de convolution. Nous allons maintenant discuter des diverses techniques permettant de les estimer et de les contrôler.

7.5.2 Réduction de l'erreur

Au vu de l'étude qualitative des erreurs d'approximation effectuée à la section précédente, notre politique de réduction de l'erreur sera basée sur la *subdivision* des éléments en vue de réduire leur extension le long de D ou dans le plan orthogonal à D selon le cas. Nous calculerons alors autant de nouvelles convolutions que nous effectuerons de subdivisions.

Les convolutions calculées sont ensuite combinées d'une manière adéquate selon la nature de l'élément géométrique subdivisé. Nous expliquons dans cette section comment effectuer ces subdivisions sans nous préoccuper du choix entre ces différents modes de réduction de l'erreur, qui sera traité aux sections 7.5.4 et 7.5.5.

(a) Subdivision du récepteur

Subdiviser le récepteur consiste à partitionner le récepteur en *sous-récepteurs* et calculer séparément une texture de visibilité pour chacun d'entre eux, comme suggéré par la figure 7.24.a. Les textures de visibilité obtenues sont ensuite collées sur les zones correspondantes du récepteur. Cette opération vise à améliorer la qualité de la texture de visibilité, en prenant en compte de manière plus locale les distances relatives mesurées verticalement entre la source, les polygones de l'obstacle et les points du récepteur.

Lors de la subdivision du récepteur, il faut prêter une attention particulière au choix de la direction D utilisée pour chacun des sous-récepteurs. La figure 7.23 montre ce qui arrive dans chacun des cas suivants :

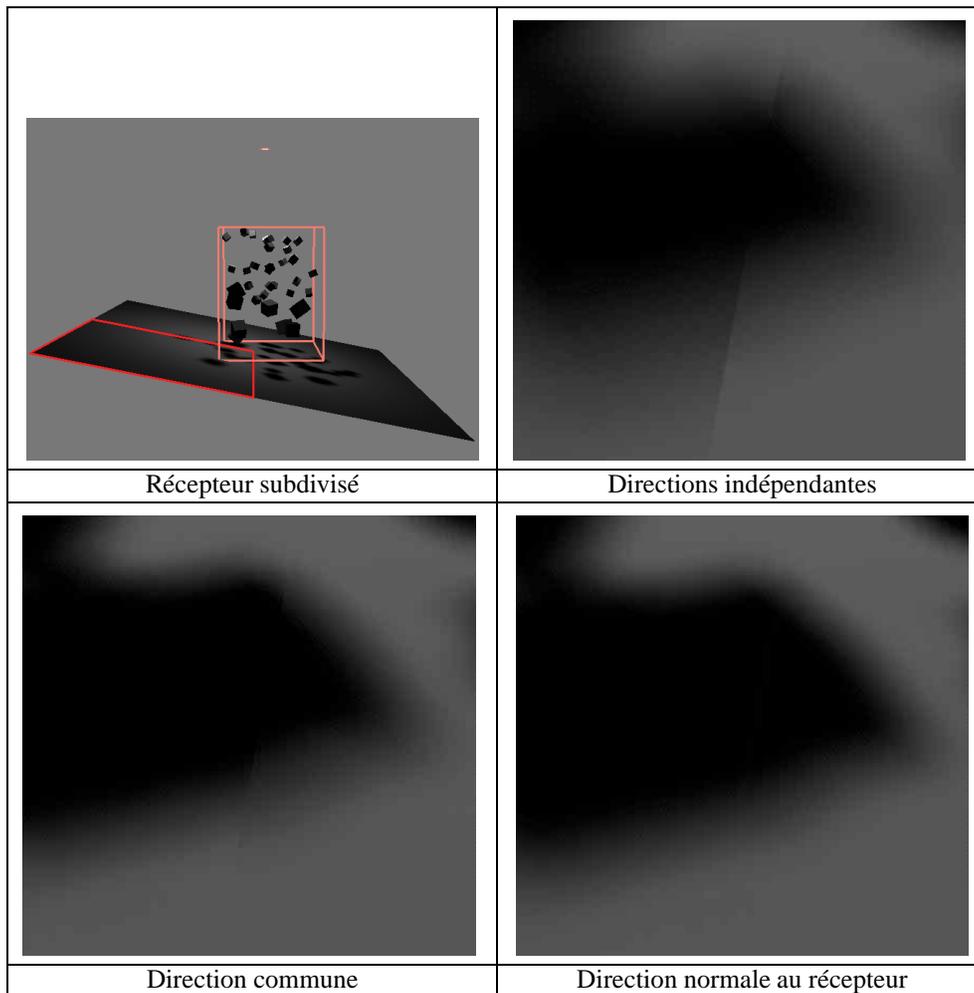


FIG. 7.23: Artefacts produits par la subdivision du récepteur selon le choix des directions D utilisées pour chaque sous-texture.

1. Si D est choisi de manière indépendante pour chacun des sous-récepteurs, la dureté des ombres change brutalement et l'on peut observer une discontinuité

2. Si D est la même pour tous les sous-récepteurs, les discontinuités sont fortement réduites.
3. Dans le cas idéal, D est choisie comme étant la normale au récepteur. Dans ce cas les pénombres calculées ont toutes même dureté et s'ajustent donc parfaitement. Bien entendu, ce choix ne correspond pas à une augmentation de la précision de la texture de visibilité, puisque les distances relatives mesurées le long de cette direction, entre la source, l'obstacle, et chacun des récepteurs sont les mêmes que celles mesurées avec le récepteur original. La texture calculée par subdivision est alors exactement celle obtenue sans subdivision.

Ces remarques confortent l'idée que la subdivision du récepteur n'a d'intérêt que si celui-ci est étendu dans la direction parallèle à D . Le temps de calcul de la texture de visibilité pour tout le récepteur augmente avec la subdivision, puisqu'il faut effectuer autant de convolutions qu'il y a de sous-récepteurs.

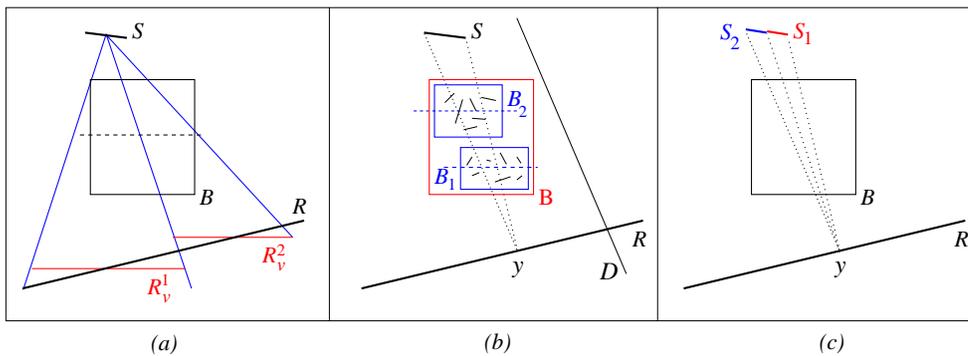


FIG. 7.24: Principe de subdivision de l'obstacle, du récepteur et de la source. (a) le récepteur est subdivisé en deux sous-récepteurs dont les récepteurs virtuels associés sont notés R_v^1 et R_v^2 . (b) l'obstacle B est subdivisé en deux sous obstacles B_1 et B_2 . Les pointillés indiquent la position de leurs plan virtuels respectifs. (c) la source est coupée en deux sous-sources

(b) Subdivision de l'obstacle

La subdivision de l'obstacle consiste à partitionner l'obstacle en sous-obstacles plus petits et à calculer une nouvelle texture de visibilité pour chaque sous obstacle pris séparément (C.f figure. 7.24.b). Les textures de visibilité sont ensuite combinées entre elles de manière à prendre en compte les relations de visibilité entre les sous-obstacles. Nous discuterons de la méthode permettant de combiner les différentes textures de visibilité obtenues, à la section 7.5.3.

Le partitionnement de l'obstacle peut être effectué de façon totalement arbitraire. Cependant, en vue de réduire l'erreur due à la trop grande extension de l'obstacle dans la direction parallèle à D , il est plus judicieux de choisir des sous obstacles occupant des intervalles d'altitudes les plus espacés possible. Dans notre implémentation, l'accès aux sous-obstacles est facilité par la représentation des objets sous la forme d'une hiérarchie de boîtes englobantes (ou *clusters*).

Lors de la subdivision de l'obstacle, nous conservons la transformée de Fourier de l'image de la source afin d'économiser de la puissance de calcul. Nous utilisons pour cela la même direction de projection D pour le calcul de la texture de visibilité correspondant à chaque sous-obstacles et nous réglons simultanément les paramètres de mise à l'échelle des images de la source et de chacun des sous-obstacles.

(c) Subdivision de la source

Dans un certain nombre de cas particuliers la subdivision de la source est le seul moyen efficace de réduire l'erreur d'approximation. Il existe également des configurations dans lesquelles subdiviser la source permet de corriger efficacement les artefacts visuels bien plus rapidement qu'en subdivisant l'obstacle ou le récepteur. Dans tous les cas, la source est partitionnée en sous-sources (C.f Fig. 7.24.c), et une texture de visibilité est calculée pour chaque sous-source. Au moment du rendu, les textures de visibilité sont ajoutées *après modulation* grâce aux capacités de *blending* d'**OpenGL**.

Un exemple typique de situation dans laquelle la subdivision de la source s'impose est la configuration montrée sur la figure 7.25, où l'obstacle est formé d'un objet verticalement très étendu "orthogonal" à la direction D . Dans un pareil cas, l'image de l'obstacle virtuel ne contient presque rien puisque l'obstacle se projette sur un segment. La texture de visibilité est donc vide (*En haut à droite*, sur la figure). Le remède est de subdiviser la source en la coupant par le plan support du polygone de l'obstacle. L'image calculée (*En bas à gauche*) est alors très proche de la solution de référence (*En bas à droite*).

Bien que les configurations de ce type puissent être détectées en mesurant les angles d'incidence de la direction D sur les polygones de l'obstacle, il est difficile de prendre une décision de manière automatique lorsque l'obstacle est formé de nombreux petits polygones.

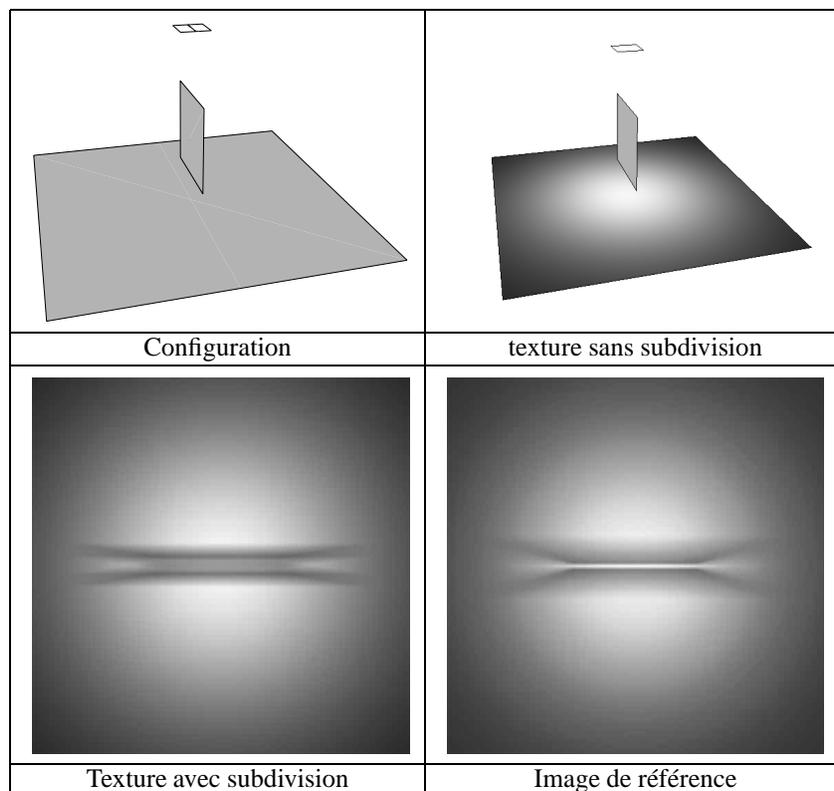


FIG. 7.25: Cas particulier où la subdivision de la source est nécessaire

Une situation où la subdivision de la source réduit l'erreur d'approximation beaucoup plus rapidement que la subdivision de l'obstacle est montrée de la figure 7.26. Dans cette situation, l'intervalle des valeurs possibles de α pour différents polygones de l'obstacle est très grand ($\alpha = 0$ pour le pied du palmier à $\alpha = 1.0$ pour les polygones du haut). La source étant très étendue et non uniforme, la corrélation entre la visibilité à travers les polygones et la source est très forte. Pour réduire cette corrélation, il suffit soit de subdiviser la source,

soit l'obstacle. Ici la source étant formée de quatre sources plus petites, l'image est bien meilleure après une seule subdivision de la source.

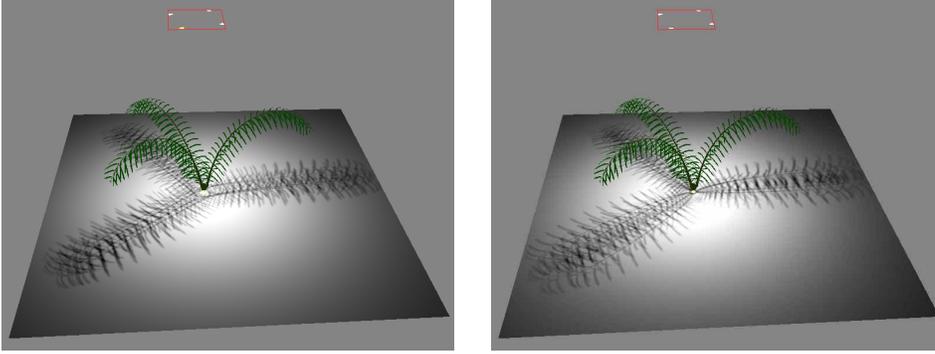


FIG. 7.26: Situation dans laquelle la subdivision de la source réduit plus rapidement les artefacts d'approximation que la subdivision de l'obstacle. Dans l'image de droite, la corrélation entre la visibilité et la source a été prise en compte.

A part dans les cas cités ci-dessus, la subdivision du récepteur et de l'obstacle suffisent à réduire efficacement les erreurs d'approximation.

7.5.3 Formule de combinaison

Alors que subdiviser la source ou le récepteur ne demande qu'une opération simple de composition des textures calculées séparément, la subdivision de l'obstacle pose le problème de savoir calculer la proportion de la source visible d'un point du récepteur à travers l'obstacle entier en fonction des mêmes quantités calculées séparément pour chaque sous-obstacle. Comme le montre l'exemple sur la figure 7.27 dans le cas de deux sous-obstacles, les valeurs $V_1(y)$ et $V_2(y)$ des textures de visibilité calculées pour chaque sous-obstacle ne représentent pas suffisamment d'information pour calculer exactement la texture de visibilité $V(y)$ correspondant à l'obstacle entier. En subdivisant l'obstacle, nous perdons en effet l'information de corrélation entre les sous-obstacles, et connaître exactement leur corrélation reviendrait à connaître exactement $V(y)$ (Nous retrouvons exactement les mêmes problèmes de corrélation qu'à la partie II). Nous proposons néanmoins dans cette section, une méthode pour approcher efficacement $V(y)$ à partir des seules valeurs $V_1(y)$ et $V_2(y)$.

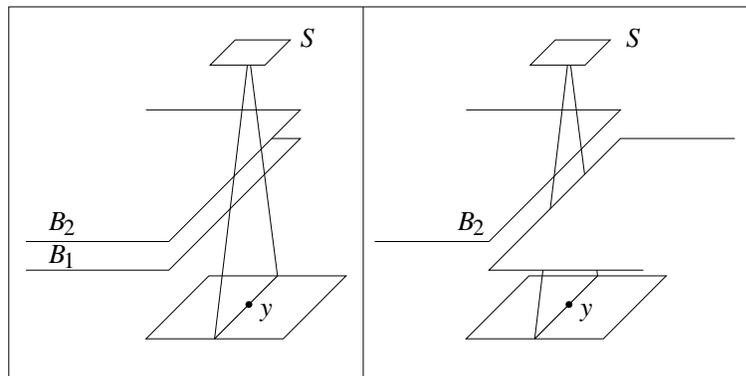


FIG. 7.27: Configuration particulière montrant qu'il est impossible de calculer exactement $V(y)$ connaissant seulement $V_1(y)$ et $V_2(y)$. Nous avons en effet dans les deux cas $V_1(y) = V_2(y) = \frac{1}{2}$, alors que, à gauche, $V(y) = \frac{1}{2}$ et, à droite, $V(y) = 0$.

(a) Principe

Nous nous limitons pour l'instant au cas de deux sous-obstacles pour lesquels nous supposons avoir séparément calculé deux textures de visibilité V_1 et V_2 . Nous proposons, connaissant $V_1(y)$ et $V_2(y)$, d'estimer $V(y)$ comme la moyenne entre les cas extrêmes correspondant à la plus forte et plus faible corrélation entre les deux sous-obstacles.

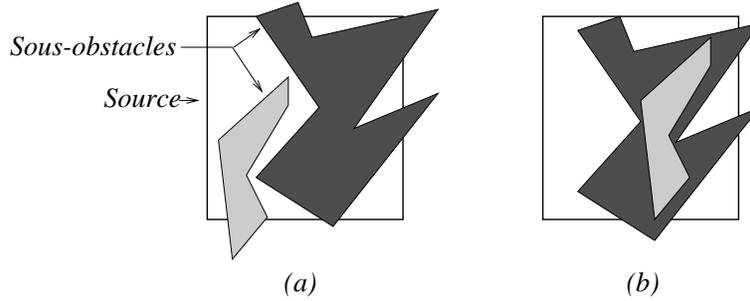


FIG. 7.28: Situations extrêmes de corrélation entre deux sous-obstacles, vu d'un point y du récepteur. En (a), les deux sous-obstacles masquent des régions disjointes de la source, en (b) la région de la source masquée par l'un est entièrement masquée par l'autre.

La corrélation entre les deux sous-obstacles est la plus faible lorsque les deux sous-obstacles masquent des parties disjointes de la source (C.f figure 7.28.a). Dans ce cas nous avons :

$$\begin{aligned} V(y) &= \max(0, V_1(y) - (1 - V_2(y))) \\ &= \max(0, V_1(y) + V_2(y) - 1) \end{aligned}$$

La corrélation entre les sous-obstacles est la plus forte lorsque, vue d'un point y sur le récepteur, la partie de la source masquée par un sous-obstacle est entièrement masquée par l'autre (C.f figure 7.28.b). Dans ce cas, nous avons :

$$V(y) = \min(V_1(y), V_2(y))$$

Nous choisissons donc la valeur suivante pour approcher la proportion de la source masquée par l'obstacle entier :

$$V(y) = \frac{1}{2} (\min(V_1(y), V_2(y)), \max(0, V_1(y) + V_2(y) - 1)) \quad (7.3)$$

L'erreur d'approximation attachée à cette formule de composition est maximale dans le cas montré sur la figure 7.27 où elle vaut $\frac{1}{4}$. Une telle situation se produit cependant rarement. Il faut pour cela que deux polygones ayant deux arêtes parallèles soient traités comme sous-obstacles. Les points où l'erreur est maximale sont ceux appartenant au plan défini par ces deux arêtes. L'effet de cette erreur est visible sur l'avant dernière image de la figure 7.31 ($\epsilon = 0.0$, 210 sous-clusters) : pour certains cubes la pénombre a un gradient différent de la solution de référence. En effet, des faces adjacentes d'un mêmes cube appartiennent à des sous-obstacles différents. Cette erreur est donc plutôt un artefact de la mauvaise qualité des boîtes englobantes formant la hiérarchie des obstacles. Le reste du temps, l'erreur d'approximation contenue dans la formule de combinaison est faible. Elle est notamment nulle partout où l'une des deux valeurs $V_1(y)$ et $V_2(y)$ vaut 0 ou 1.

Enfin, cette formule s'étend facilement au cas de plusieurs sous-obstacles. Nous les combinons pour cela deux par deux récursivement jusqu'à n'obtenir qu'une seule texture de visibilité.

(b) Implémentation

Comme nous l'avons vu, les images de convolution calculées pour chaque sous-obstacle n'ont pas nécessairement la même échelle ni le même centre. Pour chaque point de l'image combinée il faut donc effectuer un changement de repère pour connaître le point correspondant dans chacune des textures calculées. La valeur de ces textures est alors estimée par interpolation bilinéaire sur les valeurs des pixels les plus proches du point trouvé. La combinaison des valeurs de ces pixels correspond ensuite à un simple appel de fonction réalisant le calcul du membre de droite de (7.3).

Bien que nous ne l'ayons pas testée, nous proposons une implémentation plus efficace basée sur l'utilisation directe de la carte graphique : l'expression (7.3) peut en effet se ramener à un certain nombre d'opérations simples que l'on peut effectuer sur des images grâce aux fonctionnalités de la librairie **OpenGL** : somme, différence, produit par une valeur uniforme, minimum et maximum de deux images, qui sont disponibles sous primitives ou extensions. Les changements de repères nécessaires à l'ajustement des textures de visibilité avant combinaison sont alors effectués automatiquement au moment de l'affichage des textures.

N.B : On peut se passer des extensions *EXT_blend_minmax* servant à calculer le minimum et le maximum en utilisant le fait que **OpenGL** restreint les valeurs des pixels à $[0, 1]$ et calcule donc implicitement en chaque pixel $\min(\max(0, I(y)), 1)$ à l'affichage d'une image I . Nous écrivons pour cela l'égalité suivante, qui donne la suite des opérations à effectuer :

$$\min(v_1, v_2) = \frac{1}{2}v_1 + \frac{1}{2}v_2 - \frac{1}{2}(\max(v_1 - v_2, 0) + \max(v_2 - v_1, 0))$$

7.5.4 Estimation perceptive de l'erreur d'approximation

Pour estimer l'erreur d'approximation dans les textures de visibilité que nous calculons par convolution, nous avons besoin d'estimer la différence entre la texture de visibilité calculée et la visibilité exacte. La méthode la plus simple pour mesurer la différence entre deux images I et I' est d'utiliser une norme usuelle, par exemple la norme L_1 :

$$E(I_1, I_2) = \int |I(x) - I'(x)| dx$$

Nous obtenons alors une borne sur la distance mathématique qui sépare les deux images. Cette méthode n'est cependant pas du tout adaptée à l'estimation de l'erreur pour notre méthode pour les raisons suivantes, que nous avons déjà évoquées au chapitre 2 :

- De telles bornes sont globales et manquent donc les phénomènes d'erreur locale ;
- elles ne caractérisent pas l'erreur de la même façon qu'un œil humain : Considérons par exemple deux images différant d'un pixel. Leur distance mesurée en utilisant la norme $\|\cdot\|_\infty$ peut alors être très grande bien que ces images restent semblables à l'œil. Des exemples équivalents peuvent ainsi être trouvés pour les autres normes usuelles [SD95] ;
- dans le but d'obtenir un algorithme conduit par l'erreur d'approximation, nous avons besoin de connaître une relation simple entre l'influence des approximations géométriques réalisées et l'erreur ainsi estimée pour pouvoir agir rétroactivement sur ces approximations. Les bornes obtenues par majoration de la distance entre deux textures de visibilité au sens des normes usuelles dépendent de manière non triviale des paramètres géométriques de la scène (Taille de l'obstacle, coefficient α, \dots).

Nous proposons donc de construire une mesure *perceptive* de l'erreur, *i.e* estimant l'erreur perçue par l'œil. En termes d'artefacts visuels l'erreur d'approximation est essentiellement caractérisée par le fait que la dimension des zones de pénombre produites par les

différents polygones formant l'obstacle dépend fortement des distances relatives entre ces polygones, la source et les points du récepteur. En revanche, dans la texture de visibilité calculée, toutes les zones de pénombre ont même *dureté*. Nous choisissons donc d'estimer l'erreur d'approximation associée à l'usage de la géométrie virtuelle par l'amplitude de la dimension des zones de pénombre sur le récepteur.

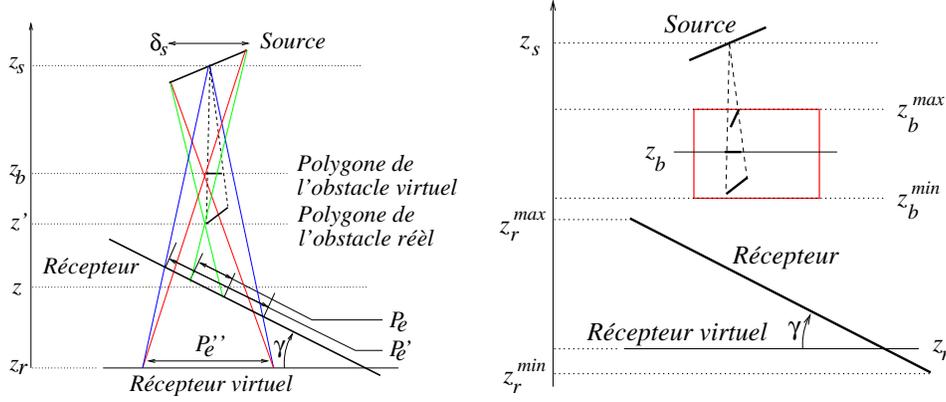


FIG. 7.29: Notations pour la construction du critère d'erreur.

Nous allons présenter les calculs sur la base de schémas en deux dimensions. Ce calcul se généralise aisément en trois dimensions en remplaçant les longueurs mesurées par les diamètres des objets correspondants. Nous adoptons d'autre part les notations de la partie gauche de la figure 7.29.

Considérons un polygone de l'obstacle, et notons z' l'altitude minimum de ses points. Soit z_b l'altitude de l'obstacle virtuel. La taille de la zone de pénombre produite par ce polygone vaut donc approximativement au niveau d'un point d'altitude z du récepteur (C.f figure 7.29) :

$$P_e'(z) = \frac{1}{\cos \gamma} \frac{z' - z}{z_s - z'} \quad (7.4)$$

D'autre part, la dimension de la zone de pénombre causée par l'obstacle virtuel sur le récepteur virtuel est donnée par :

$$P_e'' = \alpha \delta_s \quad \text{avec} \quad \alpha = \frac{z_b - z_r}{z_s - z_r}$$

Après projection sur le récepteur réel, la dimension de la pénombre calculée devient :

$$P_e(z', z) = \delta_s \frac{1}{\cos \gamma} \frac{z_s - z}{z_s - z_r}$$

L'erreur relative sur la dimension de la zone de pénombre calculée vaut donc :

$$\Delta_e(z', z) = \frac{P_e'(z)}{P_e(z', z)} = \frac{(z_s - z')(z_s - z)}{z' - z} \times \frac{z_b - z_r}{(z_s - z_r)(z_s - z_b)}$$

Notons maintenant z_b^{\min} et z_b^{\max} les altitudes minimale et maximale des objets constituant l'obstacle, z_r^{\min} et z_r^{\max} l'altitude minimale et maximale des points du récepteur. Sans perte de généralité¹, nous supposons que

$$z_r^{\max} < z_b^{\min}$$

¹Cette inégalité est toujours vérifiée en restreignant le récepteur à la partie sur laquelle se projettent les ombres

Dans ce cas, nous avons montré que l'erreur d'approximation $\Delta_e(z', z)$ atteint son maximum lorsque $z = z_r^{max}$ et $z' = z_b^{min}$ et son minimum lorsque $z = z_r^{min}$ et $z' = z_b^{max}$. La différence entre ces deux valeurs donne l'amplitude de la variation de taille de pénombre :

$$E(R, B) = \Delta_e(z_b^{min}, z_r^{max}) - \Delta_e(z_b^{max}, z_r^{min}) \quad (7.5)$$

Contrairement aux estimateurs basés sur les normes usuelles, la mesure de l'erreur que nous venons de construire prend effectivement en compte l'erreur telle qu'elle est perçue par l'œil humain. Il est de plus facile de vérifier que sa valeur décroît lorsque les éléments tendent vers la géométrie virtuelle, *i.e* lorsque le récepteur et l'obstacle sont plans et parallèles à la source. Plus précisément, lorsque B_1 (*resp.* R_1) est un sous-obstacle (*resp.* sous-récepteur) de B (*resp.* R) au sens de l'inclusion, nous avons

$$E(R_1, B) \leq E(R, B) \quad \text{et} \quad E(R, B_1) \leq E(R, B) \quad (7.6)$$

Finalement, ce critère d'erreur dépend de façon simple et intuitive de la géométrie des éléments. Nous l'avons utilisé pour construire l'algorithme de contrôle de précision dans les textures de visibilité que nous décrivons dans la section suivante.

7.5.5 Algorithme de contrôle de l'erreur

Pour construire un algorithme de contrôle de l'erreur, nous avons utilisé les méthodes de réduction de l'erreur vues en 7.5.2 en coordination avec le critère d'erreur perceptif introduit en 7.5.4. Après en avoir décrit le principe, nous discuterons de sa convergence et en donnerons des exemples d'utilisation.

(a) Principe

Pour une configuration source/obstacle/récepteur donnée le critère d'erreur permet d'estimer si la précision de la texture de visibilité est suffisante en comparant $E(R, B)$ à une limite E_{max} fixée par l'utilisateur. Si la précision n'est pas suffisante, le calcul est effectué de manière récursive en subdivisant le récepteur ou l'obstacle, comme expliqué en 7.5.2.(b) et 7.5.2.(a). Nous calculons pour cela l'erreur obtenue après subdivision en fonction des dimensions des sous-éléments utilisés et nous choisissons l'alternative conduisant à la meilleure précision. L'erreur en cas de subdivision du récepteur vaut :

$$E(R_1, R_2, B) = \max(E(R_1, B), E(R_2, B))$$

En revanche, si l'on subdivise l'obstacle, elle s'écrit :

$$E(R, B_1, B_2) = C_{err} + \max(E(R, B_1), E(R, B_2))$$

Dans cette expression, la quantité C_{err} représente l'erreur d'approximation de la corrélation associée à la formule de combinaison (*C.f* section 7.5.3). Cette erreur est dans une large mesure négligeable.

Il est important de noter que l'estimation de l'erreur est de type *a priori*, ce qui nous permet de connaître à l'avance l'erreur produite par une option particulière de raffinement sans avoir à effectuer explicitement le calcul de la texture de visibilité. Le choix de la méthode offrant la plus forte réduction de l'erreur à chaque étage range d'autre part notre algorithme dans la catégorie des algorithmes *gloutons*. Il est raisonnable de penser qu'une exploration plus profonde des diverses options de subdivision puisse conduire à une meilleure réduction globale de l'erreur. Nous n'avons pas expérimenté de telles méthodes, bien qu'elles ne posent pas de problèmes algorithmiques particuliers : nous pourrions envisager une exploration systématique des options de subdivision jusqu'à une profondeur donnée, ou mieux, allonger des chemins de subdivisions successives rangés dans une file d'attente avec priorité pondérée par le rapport coût/qualité de chaque option.

Notre version, plus simple, de l'algorithme de contrôle de l'erreur est détaillée sur la figure 7.30. La procédure *PasteTextures* colle l'une à coté de l'autre les deux sous-textures, alors que *CombineTextures* réalise leur combinaison dans le cas de la subdivision de l'obstacle.

```

Texture ComputeTexture( $S, B, R$ )
  if  $E_{max}(S, R, B) < \epsilon_{max}$ 
    return ConvolutionTexture( $S, B, R$ )
  else
    if  $E(R_1, \dots, R_n, B) < E(R, B_1, \dots, B_p)$ 
       $T_1 = \text{ComputeTexture}(S, B, R_1)$ 
      ...
       $T_n = \text{ComputeTexture}(S, B, R_n)$ 
      return PasteTexture( $T_1, \dots, T_n$ )
    else
       $T_1 = \text{ComputeTexture}(S, B_1, R)$ 
      ...
       $T_p = \text{ComputeTexture}(S, B_p, R)$ 
      return CombineTextures( $T_1, \dots, T_p$ )

```

FIG. 7.30: Algorithme de contrôle de l'erreur dans le calcul des ombres par convolution.

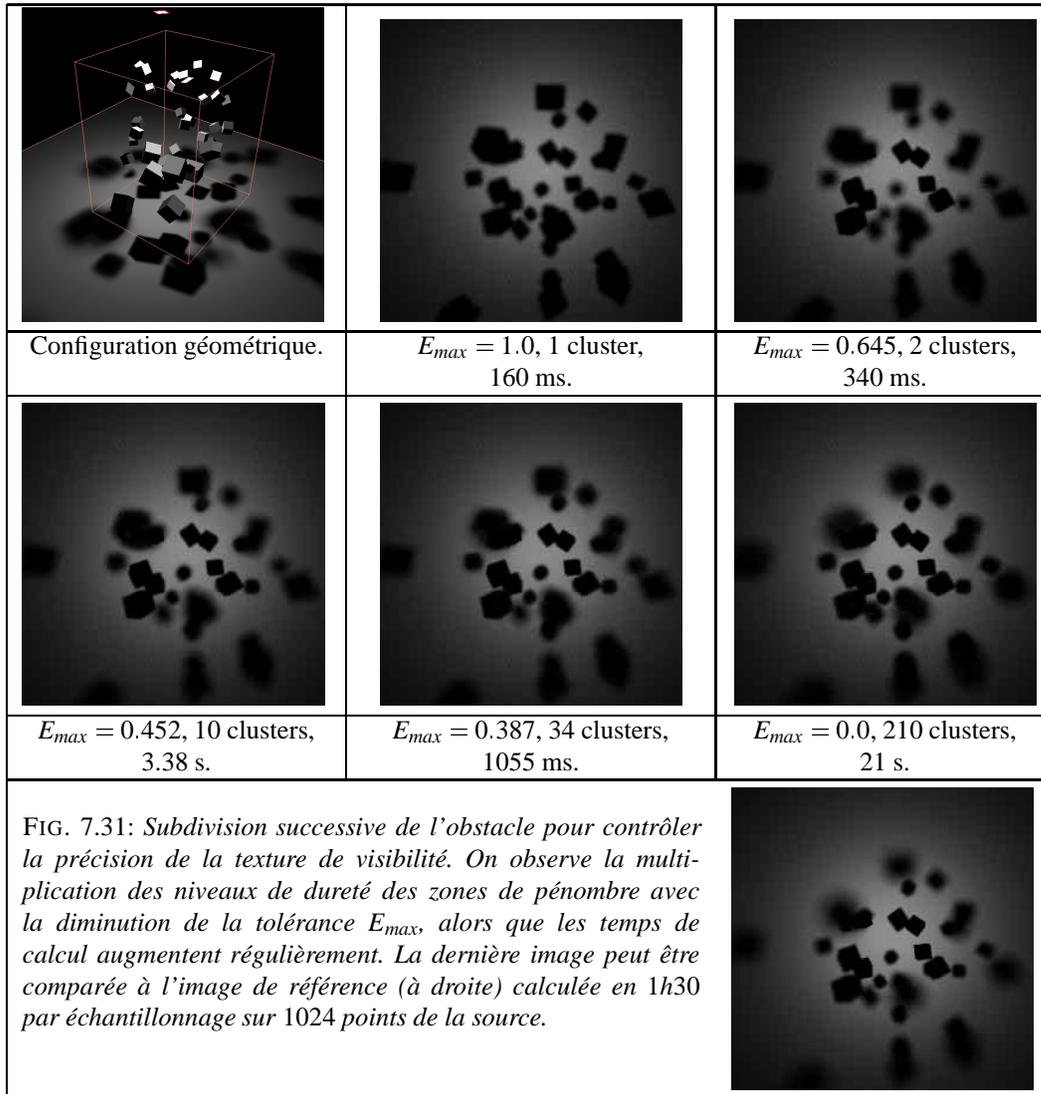
(b) Convergence

Nous nous intéressons à la question de savoir si en diminuant la tolérance d'erreur E_{max} , la texture de visibilité calculée converge effectivement vers une image représentant une carte de la visibilité exacte sur le récepteur. La réponse est clairement négative pour plusieurs raisons :

Tout d'abord, lorsque la source est un polygone, la direction de projection D n'est pas nécessairement normale à la source dont l'extension verticale n'est donc pas prise en compte dans le critère d'erreur. Cette remarque tient lorsque la source n'est pas non plus plane.

Ensuite, il n'est guère possible de subdiviser la hiérarchie des objets constituant l'obstacle plus loin qu'à un niveau où chaque polygone est traité séparément. En effet, bien qu'il soit techniquement possible de découper l'obstacle en tranches horizontales aussi fines que possible, les erreurs d'approximation de la méthode de combinaison finissent pas devenir significatives, notamment à cause de la situation de corrélation particulière entre les tranches. Nous nous limitons donc en général à la subdivision naturellement permise par la hiérarchie de clusters qui constitue l'obstacle, limite avant laquelle C_{err} reste négligeable. La taille δ_z (mesurée verticalement) du plus grand polygone de l'obstacle permet donc, en utilisant l'équation (7.5), de prévoir l'erreur minimale dans la texture de visibilité calculée.

Notre algorithme de contrôle de l'erreur est cependant tout à fait utilisable en pratique car en subdivisant le récepteur ou l'obstacle, le critère d'erreur diminue de manière significative (Voir la relation 7.6), et ce de façon régulière au fur et à mesure de la subdivision. Cela signifie que la quantité de calculs nécessaires à atteindre une précision fixée croît régulièrement avec cette précision. D'autre part, comme le montre la figure 7.31, les textures de visibilité obtenues pour une précision maximale sont quasiment indiscernables à l'œil nu des images de référence. Cet algorithme répond donc exactement à notre attente concernant le contrôle de l'erreur.



7.6 Résultats

Nous présentons dans cette section divers exemples d'utilisation de la méthode de convolution pour simuler des ombres progressives. Dans ces exemples, l'éclairage sur les polygones recevant des ombres est calculé par la méthode de convolution alors que les autres polygones (par exemple les obstacles eux-mêmes) sont éclairés par la méthode de radiosité.

La scène présentée sur la figure 7.32 contient un obstacle formé de 45000 polygones (le mobile). Malgré cette apparente complexité, les temps de calculs sur l'*Onyx*² sont très faibles, puisque la partie du calcul traitant la géométrie est effectuée par le moteur de rendu graphique, très performant, de l'*Onyx*².

Pour plus de précision, de plus en plus de convolutions sont utilisées dans la synthèse des ombres : dans la dernière image, l'ombre de l'oiseau placé contre le mur est plus nette, du fait de sa faible distance au mur. Les temps figurant sous chaque image correspondent respectivement aux temps de calculs des trois textures (une sur chaque mur et une troisième sur le sol) pour des tailles respectives de 128² and 256² pixels.

La figure 7.33.a montre un exemple d'effet de bord intéressant automatiquement pris

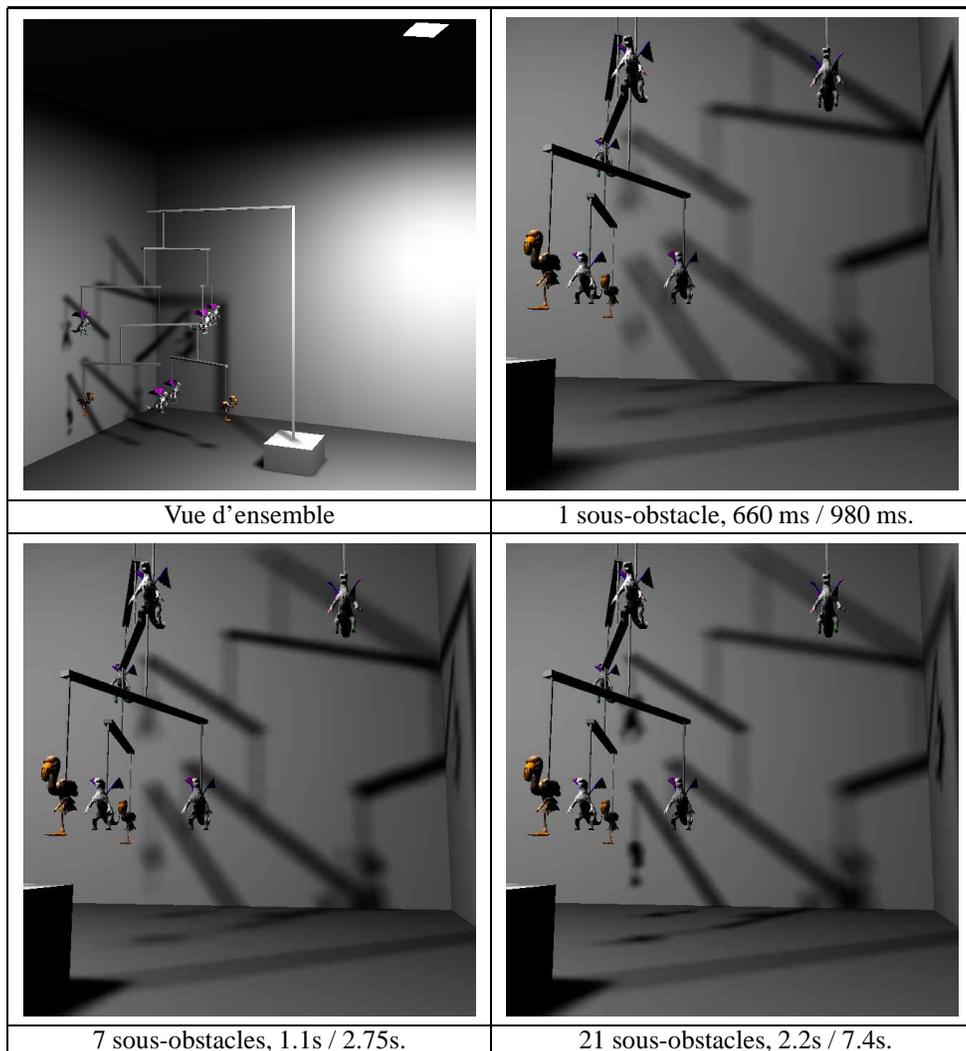


FIG. 7.32: Exemple de contrôle de l'erreur par raffinement de l'obstacle (voir texte). Le mobile est un obstacle formé de 45000 polygones. Les temps sont ceux du calcul des trois textures (murs et sol) pour les tailles 128^2 et 256^2 pixels sur une *Onyx²Infinite Reality*.

en charge par notre méthode : la source est un volume constitué de 360 polygones. Elle éclaire le mur du fond à travers un trou triangulaire dans un écran. La convolution simule automatiquement le renversement de l'image de la source, comme le ferait un projecteur réel.

Sur la figure 7.33.b, nous voyons un autre exemple de calcul d'ombre progressive pour un obstacle complexe. Une unique convolution a servi à calculer la texture de visibilité projetée sur l'ensemble des cubes.

Le tableau sur la figure 7.34 donne les temps de calcul pour les scènes *cubes* de la figure 7.31, *pyramide* de la partie droite de la figure 7.33, et *mobile* de la figure 7.32. Lorsque la résolution des textures augmente, les temps de calcul sont largement dominés par le calcul des Transformations de Fourier Rapides.

Ces résultats démontrent l'efficacité de la méthode de convolution par rapport aux méthodes classiques d'échantillonnage de la source, notamment par l'absence d'artefacts dans les ombres produites. Dans notre méthode, la source est échantillonnée au sens où elle est transformée en une image discrète, mais nous traitons la contribution de tous les échan-

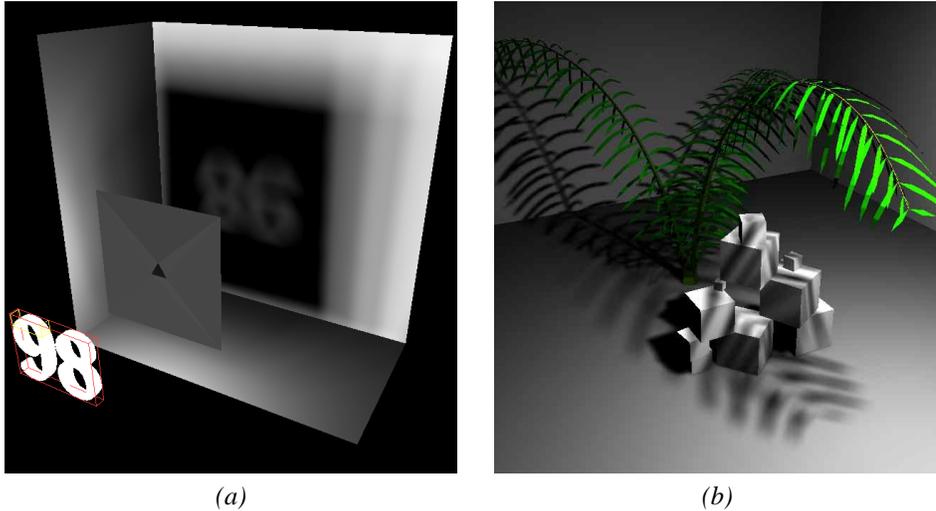


FIG. 7.33: Exemples d'utilisation de la méthode de convolution. A gauche : en une seule convolution, la méthode a simulé le renversement de l'image de la source volumique (360 polygones) après son passage à travers un petit trou triangulaire. A droite : Une unique texture de visibilité de taille 128×128 a été calculée pour l'ensemble des cubes, ce qui constitue un exemple de récepteur non polygonal.

Scène	FFT seule		Cubes 212 polys		Pyramide 4340 polys		Mobile 45000 polys	
	O_{nyx}^2	O_2	O_{nyx}^2	O_2	O_{nyx}^2	O_2	O_{nyx}^2	O_2
Machine	6	10	50		60		220	
128×128	6	10	50		60		220	
256×256	25	40	170		170		340	
512×512	110	170	610	1100	620	1190	770	
1024×1024	500	770	2510	5440	2540	4980	2730	

FIG. 7.34: Temps de calcul associés au calcul d'une texture (sol ou mur) des scènes Cubes, Pyramide et Mobile

tillons d'un seul coup. Même sans subdivision, la méthode de convolution est capable de produire des ombres progressives visuellement plausibles, sans discontinuités. L'erreur est en effet distribuée de telle manière que l'œil ne la remarque pas. Enfin, lorsqu'une texture de visibilité est calculée avec subdivision de l'obstacle, chaque polygone de l'obstacle est pris en compte exactement une fois. Seule change la spécificité des paramètres qui sont utilisés pour rendre son ombre.

7.7 Conclusion

Nous avons présenté dans ce chapitre une méthode de calcul des ombres progressives basée sur le calcul d'une convolution entre des images convenablement choisies de la source et d'un obstacle. Cette méthode rend des calculs géométriques, ordinairement complexes, accessibles sous la forme d'une approximation basée sur un calcul de transformée de Fourier. Ce transfert de besoins en termes de puissance de calcul s'inscrit naturellement dans l'évolution de la technologie informatique : de plus en plus de machines sont aujourd'hui équipées de composants dédiés au traitement du signal (Stations O_2 , technologie MMX), desquels la méthode de convolution tire sa rapidité. Cette méthode est également hautement parallélisable, non seulement par le calcul des Transformées de Fourier, mais également par les algorithmes de contrôle de la précision que nous avons présentés.

Notre méthode s'est révélée expérimentalement très robuste et d'une grande facilité d'automatisation. Nous pensons qu'elle pourrait trouver des applications dans tous les domaines où le calcul d'ombres complexes à des vitesses interactives est nécessaire : simulation de l'éclairage en architecture (par exemple à des fins commerciales), réalité virtuelle, jeux vidéo mais également des simulations botaniques, pour lesquelles elle permettrait une prise en compte rétroactive de l'ombrage du couvert végétal sur lui même dans un algorithme de simulation physiologique de la croissance des plantes.

Grâce au contrôle de la précision, elle permet de plus de régler la qualité du calcul en fonction du temps imparti. Nous utiliserons cette capacité dans le chapitre suivant pour développer l'intégration de cette méthode dans un algorithme de radiosité hiérarchique. Nous verrons que l'algorithme obtenu nous permet d'obtenir en des temps très brefs des solutions de radiosité très fines dans des environnements complexes.



Les algorithmes de simulation de l'éclairage utilisent généralement une unique notion de qualité pour conduire le contrôle de la précision du calcul. Nous avons cependant vu qu'une estimation purement quantitative de l'erreur sur une solution de radiosit  ne permet pas de contr ler les artefacts visuels, de m me qu'une m thode utilisant une approche perceptive de l'erreur ne renseigne pas sur la distance d'un r sultat   la solution cherch e. Utiliser une m me m thode de contr le de la pr cision pour des types d'erreur de diff rentes natures peut   l'extr me, conduire   des incoh rences: subdiviser tr s finement le maillage d'une sc ne pour simuler une ombre particuli re alourdit  norm ment la r solution de l' quation discr te de radiosit , mais ne r duit qu'un peu l'erreur sur la balance  nerg tique globale. Nous allons pr senter dans une derni re partie, comme synth se de nos travaux, un algorithme de calcul contr l  de la radiosit  dans lequel cohabitent plusieurs notions de qualit . Dans cet algorithme, le calcul de l' quilibre global des  changes  nerg tiques d'une part, et la simulation de d tails d'ombre tr s fins d'autre part, font en effet s par ment l'objet de m thodes sp cifiques.

Quatrième partie

Synthèse

Chapitre 8

Radiosité hiérarchique avec convolution

E EN DÉPIT des récents progrès des techniques de rendu, et particulièrement des méthodes hiérarchiques de radiosité, le calcul d'une simulation précise, et notamment incluant des détails d'ombres portées très fins, reste une opération coûteuse.

Les méthodes hiérarchiques de radiosité [HSA91, SAG94, Sil95] ont en effet apporté à la simulation de l'éclairage la capacité d'échanger de façon contrôlable du temps de calcul en précision : les transferts d'énergie sont calculés au niveau le plus élevé possible tout en assurant sur le calcul un degré de précision fixé par l'utilisateur. On économise ainsi les calculs de facteurs de forme et donc de visibilité qui ne sont pas rendus nécessaires par la précision demandée par l'utilisateur.

Toutefois, les méthodes hiérarchiques de radiosité sont essentiellement utilisées pour leur capacité à fournir rapidement des approximations de la solution cherchée. Elles restent en effet très coûteuses pour le calcul de solutions de haute qualité. Selon les applications, les critères de qualité sont cependant très variables : dans le cas de la simulation de l'éclairage en architecture, par exemple, une grossière évaluation de l'équilibre global des échanges lumineux suffit. En revanche, dans les simulations se voulant plus réalistes visuellement, une grande importance est accordée aux détails et notamment aux ombres les plus fines [MK94].

De tels besoins poussent les méthodes hiérarchiques de radiosité dans un domaine pour lequel elles ne sont plus adaptées : une solution précise demande en effet un raffinement poussé et donc un grand nombre de liens. Les calculs de facteurs de forme pour des sous-liens sont d'autre part réalisés indépendamment alors qu'il existe une forte corrélation de visibilité entre les liens fils d'un lien donné. De plus, pousser le raffinement pour tenir compte de la visibilité, c'est à dire pour simuler des ombres fines, oblige à re-calculer le noyau de la radiosité qui, lui, ne nécessite souvent pas de raffinement du fait de sa plus faible variation. L'équilibre global de l'énergie pourrait dans de nombreux cas être calculé à un niveau bien plus élevé que celui où on évalue localement la visibilité afin de produire ces ombres. Le temps de rendu est donc inutilement augmenté par rapport à l'information ajoutée.

Dans ce chapitre, nous proposons de dé-corréler le calcul de la visibilité de celui de l'éclairage direct en effectuant un traitement spécial pour les détails d'ombre très fins, tout en calculant l'équilibre énergétique au niveau qui lui convient. Pour y parvenir, les transferts d'énergie incluant des calculs complexes de visibilité sont effectués grâce à la méthode de convolution présentée au chapitre 7 pour lesquels elle est spécialement adaptée. La méthode de calcul des transferts d'énergie propre à la radiosité hiérarchique est réservée aux transferts à caractère plus global.

Dans notre système, une partie de la radiosité est représentée par des textures issues de

la méthode présentée au chapitre 7. L'utilisation à l'intérieur d'un algorithme de rendu, de textures représentant de la radiosité, n'est pas nouvelle : la représentation de la radiosité sous cette forme permet l'accélération du rendu [MK94] ou l'utilisation ultérieure de la radiosité dans une passe future de la simulation [Arv86, Hec90].

Comme nous l'avons dit, nous allons donc tirer partie des avantages de la méthode de convolution pour le calcul des ombres : cette méthode simule en effet naturellement des détails d'ombre très fins et produit des ombres progressives sans artefacts visuels. Elle permet de plus un contrôle de la précision du calcul grâce à l'algorithme de raffinement qui lui est propre.

Bien qu'elle permette de générer des ombres à des vitesses interactives, la méthode de convolution a un coût non négligeable et doit donc être réservée aux situations appropriées. Nous allons montrer dans ce chapitre que la méthode de convolution permet d'accélérer de façon importante la méthode de radiosité hiérarchique. Nous expliquerons également comment sélectionner les configurations dans lesquelles la méthode de convolution peut être avantageusement utilisée, comment choisir ses différents paramètres automatiquement, comment intégrer la représentation *texturée* et *uniforme* de la radiosité en un algorithme cohérent de radiosité hiérarchique et comment élaborer une méthode de contrôle de la précision intégrant les différentes techniques de raffinement.

La méthode que nous allons décrire permet de calculer des simulations d'équilibre des échanges lumineux incluant des ombres très fines en quelques secondes. Ce travail a été publié au 9th *Eurographics Workshop on Rendering* [SS98a].

Nous commencerons tout d'abord par motiver l'utilisation de la méthode de convolution dans un algorithme de radiosité hiérarchique avant d'en exposer le principe.

8.1 Motivation

D'après les mesures présentées au chapitre précédent, le temps de calcul d'une texture de convolution est inférieur à 400ms pour une résolution de 256×256 sur *Onyx*² et *O₂*, et pour des obstacles dont le nombre de polygones atteint 40000. D'une manière générale, alors que les textures de grande taille (256×256 ou plus) ne sont nécessaires que pour les très grands polygones (murs, sol), une texture de taille 64×64 suffit pour la plupart des récepteurs.

Par ailleurs, dans la méthode de radiosité hiérarchique, la prise en compte de détails, et notamment des ombres progressives, impose une forte subdivision des surfaces réceptrices, et donc un nombre très important de facteurs de forme à évaluer comprenant des tests de visibilité coûteux. D'autre part, ces tests de visibilité sont effectués de façon indépendante, alors qu'il existe généralement une forte corrélation de la visibilité sur deux surfaces voisines.

Ces arguments suggèrent qu'utiliser la méthode de convolution pour calculer ce type de transferts d'énergie permettrait de substituer très avantageusement une simple texture de radiosité à un raffinement poussé.

D'autre part, le facteur d'accélération du calcul de la solution, l'usage de textures pour représenter la radiosité sur de grands polygones est un facteur d'accélération du rendu : sur les machines les plus récentes, le temps d'affichage d'un polygone ne dépend pas du fait qu'il soit texturé ou non. Un unique polygone texturé remplace donc très avantageusement un maillage complexe.

La figure 8.1 illustre une situation dans laquelle une simple convolution remplace avantageusement un maillage fin. Le résultat du calcul est visible sur la partie inférieure de la figure 8.1. Le facteur d'accélération est ici d'environ 24 fois, pour une qualité supérieure.

Nous allons maintenant discuter des problèmes essentiels relatifs à l'utilisation de la méthode de convolution dans un algorithme de radiosité hiérarchique.

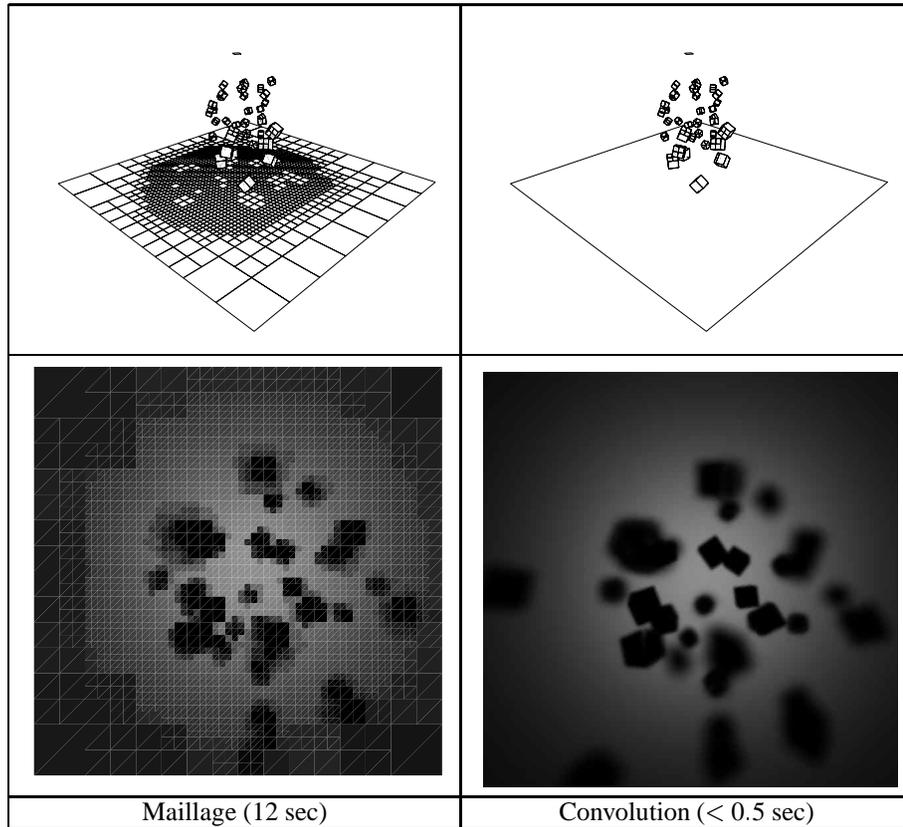


FIG. 8.1: Exemple de configuration dans laquelle une hiérarchie complexe de liens aboutissant à un maillage fin (à gauche) peut être remplacé par une unique convolution (à droite). Les images représentent le polygone formant le sol vu de dessus après calcul.

8.2 Principe

Comme nous l'avons vu au chapitre 7, la méthode de convolution doit permettre, dans le cas où la visibilité impose un raffinement poussé du récepteur, d'effectuer les calculs à moindre coût. Il serait cependant tout à fait inefficace de calculer tous les transferts d'énergie par cette méthode, notamment en l'absence d'obstacle.

Pour cette raison, nous avons construit une méthode de radiosité hiérarchique¹ admettant deux types de liens : les *liens standards*, pour lesquels le transfert d'énergie est calculé par produit entre le facteur de forme et la radiosité de l'émetteur, et les *liens de convolution*, pour lesquels le transfert d'énergie est calculé par la méthode de convolution.

Alors que les liens standards produisent des valeurs de radiosité uniformes sur la surface réceptrice, les liens de convolution produisent des textures. Il faut donc modifier l'algorithme de radiosité hiérarchique pour assurer la cohérence de la représentation et de la propagation de la radiosité sous ces deux formes. Cela signifie donc adapter les algorithmes de *gather* et de *pushpull*, mais également l'algorithme de rendu.

Disposant de deux types de liens pour représenter des transferts d'énergie, l'algorithme de raffinement doit également être modifié pour prendre en compte les nouvelles possibilités : il doit estimer l'erreur associée à chaque type de lien et savoir décider lequel est le mieux adapté à une configuration donnée. La méthode de convolution disposant de son propre algorithme de raffinement, le raffineur doit donc intégrer de manière cohérente cette dernière possibilité.

¹Ce type de méthode a été défini au chapitre 1.

Nous traitons dans la section 8.3 de l'adaptation des algorithmes de *gather* et *push/pull*. Nous proposerons un nouvel algorithme de raffinement à la section 8.4, et un nouvel algorithme de rendu sera expliqué à la section 8.5.

8.3 Propagation cohérente de l'énergie

Nous allons maintenant détailler les phases de *gather* et *push/pull* de la méthode de radiosité hiérarchique responsables de la propagation de l'énergie, pour les deux types de liens utilisés (Le principe général de la radiosité hiérarchique a été introduit à la section 1.2).

8.3.1 Calcul des transferts

Pour un lien standard entre deux éléments A_i et A_j , le calcul de l'irradiance (ou *gather*) consiste simplement à multiplier le facteur de forme F_{ij} par la radiosité B_j de A_j . Nous verrons dans la section suivante comment est calculée B_j dans le cas où la radiosité sur A_j est en partie représentée par des textures.

Pour un lien de convolution, le calcul de l'irradiance consiste idéalement à appliquer la méthode de convolution en utilisant A_j comme source, A_i comme récepteur, ainsi qu'un obstacle adéquat. Comme tout lien en radiosité hiérarchique, un lien de convolution peut aboutir à un niveau quelconque de la hiérarchie, et en particulier sur un cluster. On ne calcule donc en fait que la texture de visibilité, alors que le maillage d'affichage (Voir section 7.4.4) qui complète le calcul, est formé au moment du *push/pull*. En effet, les valeurs d'éclairage direct ne peuvent pas être calculées à un niveau de la hiérarchie où l'orientation des surfaces réceptrices n'est pas connue. Les textures de visibilité calculées sont donc temporairement stockées dans une liste attachée à l'élément récepteur.

Le calcul du transfert d'énergie pour un lien de convolution fait par ailleurs appel aux diverses extensions proposées au chapitre précédent (Voir section 7.3) : l'élément source n'étant pas nécessairement une surface élémentaire, mais peut-être un cluster, il s'agit d'une *source volumique* et éventuellement non uniforme ; le récepteur est soit une surface soit un cluster, *i.e* un récepteur volumique.

Il faut également noter que, comme dans les algorithmes classiques de *clustering*, la visibilité à l'intérieur du récepteur n'est pas spécifiquement prise en compte. Nous en parlerons à la section 8.5.2.

8.3.2 *Push/pull*

La phase de *Push/Pull* dans les algorithmes hiérarchiques de radiosité sert à rétablir la cohérence entre les valeurs de radiosité (ou d'irradiance) mises à jour à différents niveaux de la hiérarchie par la phase de *gather*. Dans notre système, ce maintien de la cohérence doit s'étendre aux textures de radiosité. Nous procédons donc de la manière suivante, l'algorithme étant résumé sur la figure 8.2 :

Au cours de la descente sont accumulées les valeurs d'irradiance jusqu'au niveau des feuilles de la hiérarchie (*i.e* des surfaces élémentaires) où elles sont converties en radiosité par produit avec la réflectance. Nous profitons de ce parcours pour récolter toutes les textures de visibilité issues des liens de convolution contribuant à la radiosité de chaque feuille. A celles-ci s'ajoutent également les textures de radiosité calculées aux itérations précédentes sur des éléments nouvellement subdivisés, qu'il faut donc également transporter jusqu'au niveau des feuilles.

Au niveau de chaque feuille, l'irradiance standard cumulée est multipliée par la valeur uniforme de la réflectance. Les textures de visibilité sont, elles, équipées d'un maillage d'affichage (*C.f* section 7.4.4.(a)) de taille adaptée à la surface élémentaire traitée, et qui complète chaque texture de visibilité en une texture de radiosité.

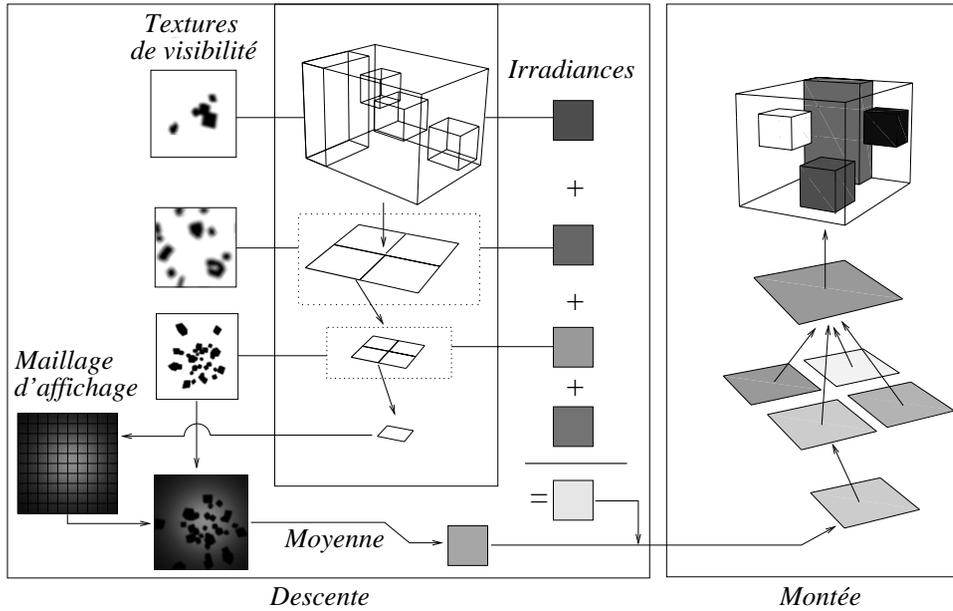


FIG. 8.2: Schéma de principe du Push/Pull.

La valeur moyenne de chaque texture de radiosité est ensuite calculée approximativement à partir des valeurs aux nœuds du maillage d'affichage, modulées par les valeurs de visibilité en ces points, récoltées dans la texture de visibilité. Cette valeur moyenne est ajoutée à la radiosité propagée durant la phase de remontée.

La radiosité est donc stockée au niveau des feuilles sous deux formes : textures et radiosité uniforme. Aux niveaux supérieurs, seule la radiosité uniforme, dont une partie provient des textures attachées aux feuilles, subsiste.

La figure 8.3 contient le pseudo-code qui correspond à l'algorithme de *Push/Pull*. Les textures de visibilité sont collectées dans une pile (variable *TextureStack*), ce qui permet de les ajouter à la liste avant la descente dans les fils de l'élément courant pour les en retirer ensuite. La procédure **PushPull** retourne la valeur moyenne de la radiosité dans la sous-hiérarchie dont la racine est l'élément *H*. Appelée sur le cluster racine de la hiérarchie, elle calcule donc la radiosité moyenne dans la scène. La fonction **ComputeDisplayMesh** calcule le maillage d'affichage et renvoie la valeur moyenne de la radiosité due à la texture.

8.4 Contrôle du raffinement

Nous décrivons dans cette section un algorithme de raffinement basé sur l'utilisation simultanée des deux représentations de la radiosité : radiosité uniforme et textures de radiosité. Après en avoir exposé le mode de fonctionnement, nous expliquerons plus en détails diverses questions soulevées par sa mise en œuvre. Nous concluons cette section par une étude expérimentale démontrant la cohérence du comportement de l'algorithme vis à vis des différents paramètres qui le caractérisent.

8.4.1 Principe

Dans notre système cohabitent deux représentations de la radiosité. Un des rôles de l'algorithme de raffinement est donc de décider, pour une configuration de transfert donnée quel type de lien employer.

```

radiosity PushPull(element H, irradiance IrradDown, stack TextureStack)
  radiosity RadUp= 0
  TextureStack→Add(H. VisibilityTextures())

  if H is a leaf
    RadUp = (H. Irradiance() + IrradDown) * H. Reflectance()
    for each T in TextureStack
      RadUp += ComputeDisplayMesh(H,T)
  else
    TextureStack→add(H. RadiosityTextures())
    radiosity R=0
    for each child c of H
      R = R + PushPull(c, IrradDown+H. Irradiance(), TextureStack)
      RadUp += R * c. Area()
    TextureStack→Suppress(H. RadiosityTextures())
    H. ClearRadiosityTextures()
    RadUp /= H. Area()

  H. Radiosity = RadUp
  TextureStack→Suppress(H. VisibilityTextures())
  H. VisibilityTextures = LISTE_VIDE
  return RadUp

```

FIG. 8.3: Pseudo-code pour l'algorithme de Push/Pull avec textures de radiosit 

Comme nous l'avons vu au chapitre pr cedent, la m thode de convolution permet, gr ce au raffinement de la source, de l'obstacle ou du r cepteur, de contr ler le degr  d'approximation dans le calcul, au prix d'un temps plus long. En termes de liens, ce raffinement sp cifique s'exprime diff remment selon l' l ment subdivis  : subdiviser la source ou le r cepteur revient   tenter d' tablir un lien   un niveau plus bas de la hi rarchie alors que subdiviser l'obstacle revient    tablir un lien au niveau courant mais pour lequel le calcul est effectu  de mani re plus pr cise. Afin de maintenir la coh rence du raffinement nous regroupons donc diff rentes possibilit s de raffinement des liens standards et des liens de convolution en trois types de subdivision :

1. subdivision de la source ;
2. subdivision du r cepteur ;
3. subdivision de l'obstacle (pour les liens de convolution uniquement).

Le raffinement d'un lien de convolution par subdivision de la source ou du r cepteur peut donc produire des liens standards et r ciproquement.  tant donn  un  metteur A_j et un r cepteur A_i , le but de l'algorithme de raffinement est de r soudre le probl me suivant :

Est-il possible d'estimer le transfert d' nergie de A_j vers A_i avec suffisamment de pr cision

- par une convolution ?
- par un lien standard ?

Si oui, quel est le type de lien le plus adapt    la configuration trait e ?

Si non, peut-on augmenter la pr cision :

- en subdivisant l' metteur ?

- en subdivisant le récepteur ?
- en subdivisant l'obstacle, si applicable ?

De même que dans un algorithme de raffinement classique, la profondeur de la subdivision doit être limitée par une variable représentant le coût maximum autorisé par l'utilisateur. La subdivision des sources ou des récepteurs est contrôlée par une limite inférieure imposée à l'aire de ces éléments ; celle des obstacles l'est par un seuil également fixé par l'utilisateur sur le coût d'évaluation d'un transfert en fonction du raffinement de l'obstacle.

8.4.2 Stratégie de raffinement

Au vu du gain de temps que représentent les liens de convolution dans les situations complexes de visibilité, notre stratégie est d'utiliser ce type de lien partout où cela est possible. Dès lors, afin de limiter l'usage des convolutions aux situations non triviales, nous imposons les conditions suivantes :

Taille minimale absolue du récepteur

Idéalement, le récepteur doit être suffisamment large pour justifier l'usage d'une texture. Ayant fixé la taille minimale d'une texture à $N_{min} \times N_{min}$, cela signifie que les polygones dont les dimensions font qu'ils n'occupent jamais à l'écran une région de plus de N_{min} pixels de large ne sont pas considérés comme récepteurs potentiels d'un lien de convolution. Un tel critère dépend cependant du point de vue particulier utilisé au moment du rendu. On fixe donc arbitrairement la taille minimale a_{min} d'un tel récepteur en proportion de la taille de la scène. En pratique, cette proportion varie entre 0.1% et 10%.

Cette condition a pour effet de limiter le nombre de textures calculées en proportion du nombre de polygones de taille suffisante, ce qui évite qu'un trop grand nombre de textures ralentisse sensiblement la vitesse d'affichage de la solution.

Présence d'obstacles

La méthode de convolution n'est intéressante que dans le cas où l'émetteur et le récepteur sont séparés par des obstacles. Étant donné une paire d'éléments A_i et A_j , il faut alors déterminer tous les obstacles susceptibles d'être pris en compte dans le calcul de la visibilité. On utilise pour cela un algorithme basé sur le *shaft-culling* [HW91a].

Un *shaft* est un volume délimitant l'ensemble des segments qui rencontrent deux boîtes aux cotés parallèles aux axes (C.f figure 8.4). Dans ce cas particulier, le *shaft* est décrit par au plus huit plans s'appuyant simultanément sur les deux boîtes auxquels s'ajoutent les faces des boîtes elles mêmes. Une fois le *shaft* construit, le test d'intersection du *shaft* avec la boîte englobante d'un objet peut être réalisé de manière très efficace [HW91a] et constitue un test exact d'intersection de l'objet avec l'ensemble des rayons reliant les deux boîtes englobantes génératrices du *shaft*.

Nous utilisons ce test pour construire un algorithme récursif de sélection de tous les obstacles potentiels entre un élément source et un élément récepteur : l'algorithme examine récursivement les fils d'un *cluster* en commençant par la racine de la hiérarchie jusqu'à éventuellement trouver des clusters rencontrant le *shaft* défini par les boîtes englobantes de l'émetteur et du récepteur, et n'ayant ni l'émetteur ni le récepteur dans la sous-hiérarchie dont il est racine. Il faut en effet éliminer les obstacles qui sont parents (plus ou moins éloignés) de l'émetteur ou du récepteur, pour éviter que ceux-ci ne projettent leur propre ombre sur le récepteur.

Par souci d'efficacité, le test de non parenté est remplacé par un test de non contenance, s'effectuant à coût constant sur les boîtes englobantes. Ce test n'est pas automatiquement équivalent au test de non parenté : on n'a en effet uniquement l'implication suivante : Si p ne contient pas q alors p n'est pas parent de q . L'implication inverse dépend de la méthode de construction des clusters et se trouve être réalisée par celle que nous employons.

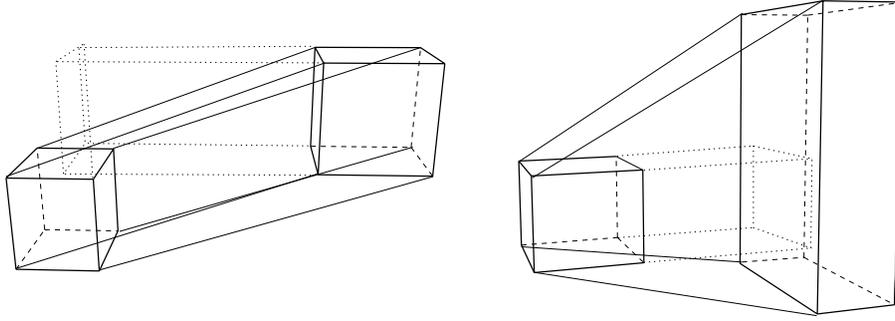


FIG. 8.4: Construction d'un shaft à quatre plans (à droite) et six plans (à gauche). Dans les configurations où les boîtes englobantes se rencontrent, le nombre de plans peut atteindre huit.

Afin de diminuer la taille des obstacles sélectionnés, un traitement spécial est appliqué à chaque obstacle afin de déterminer le nombre n de ses fils qui rencontrent effectivement le *shaft*. Si $n = 0$, l'obstacle est écarté. Si $n > 1$ l'obstacle est sélectionné. En revanche, si un seul fils rencontre le *shaft*, l'algorithme de simplification est à son tour appliqué à ce fils.

Notre algorithme de sélection des obstacles détermine en fait les obstacles potentiels, et peut donc sélectionner des obstacles ne formant pas d'ombre sur le récepteur. Cela ne pose pas de problèmes car cela est fait de façon cohérente avec la méthode de calcul des ombres utilisée : en utilisant un obstacle ainsi sélectionné (et notamment après l'élimination décrite plus haut) la méthode de convolution ne produira une ombre que si elle existe vraiment.

Finalement, l'algorithme présenté sélectionnant une liste d'obstacles, ceux-ci sont temporairement regroupés en un unique *cluster* afin de réaliser l'appel à la méthode de convolution.

Le pseudo-code décrivant l'algorithme de sélection des obstacles est présenté sur la figure 8.5.

Taille relative du récepteur

Le dernier test que nous effectuons consiste à comparer la taille relative des polygones du récepteur au gradient de l'ombre éventuellement projetée dessus. Si ce gradient est très faible, l'intensité lumineuse sur les polygones du récepteur peut-être représentée plus efficacement par des éléments standards de radiosité que l'on affiche avec interpolation linéaire (ou *smooth-shading*).

Pour cela, nous estimons le gradient Δ de la radiosité sur le récepteur en utilisant l'expression (7.4) donnant la taille approximative des zones de pénombres produites par l'obstacle. Ayant besoin d'une borne supérieure Δ_{max} du gradient, nous posons donc

$$\Delta_{max} = \frac{e_{max} \rho_{max} F_{ij}}{P'_{min}} \quad \text{avec} \quad P'_{min} = \frac{1}{\cos(\gamma_{max})} \times \frac{z'_{min} - z_{max}}{z'_s{}^{max} - z'_{min}}$$

Les quantités z'_{min} , z_{max} et $z'_s{}^{max}$ désignent respectivement les altitudes minimales et maximales d'un point de l'obstacle, du récepteur et de source le long de la direction de projection D intervenant dans la méthode de convolution. Lorsque le récepteur est un groupe de surfaces, nous utilisons $\cos(\gamma_{max}) = 1$, et lorsque $z'_{min} < z_{max}$ nous choisissons $P'_{min} = 0$.

Étant donné un obstacle constitué d'une liste de polygones, nous calculons la taille maximale T_{max} de ces polygones comme étant le plus grand diamètre d'un polygone de l'obstacle.

L'usage d'une texture pour représenter la radiosité sur le récepteur est donc intéressant

```

ListOfClusters SelectBlockers (element p, element q, cluster B, shaft s)
  if p = B or q=B
    return NULL
  if B→Contains(q) or B→Contains(p)
    ListOfClusters L = LISTE_VIDE
    For All Children b of B
      L→AddToList(SelectBlockers(p,q,b,s) )
    return L
  else
    if s→Intersects(B)
      return SimplifyBlocker(B,s)
    else
      return NULL

ListOfClusters SimplifyBlocker(cluster B, shaft s)
  integer n = 0
  cluster selected_child
  For All Child b Of B
    if s→Intersects(b)
      selected_child = b
      n = n+1
  if n = 0
    return NULL
  if n = 1
    return SimplifyBlocker(selected_child,s)
  else
    return B

```

FIG. 8.5: Algorithme de sélection des obstacles potentiels. La sélection entre un élément source S et un récepteur R s'effectue en appelant **SelectBlockers**(S , R , Root, shaft(S , R)) où Root désigne le cluster racine de la hiérarchie. La fonction **SimplifyBlocker** permet de réduire sensiblement la taille des obstacles sélectionnés (voir texte).

lorsque :

$$T_{max}\Delta_{max} \geq 1$$

Il est bien entendu que cette méthode comprend bon nombre d'approximations qui rendent le test ci-dessus un peu arbitraire. En pratique nous remplaçons donc le 1 du membre de droite par une constante fixe δ_{min} plus petite que l'unité (par exemple $\delta_{min} = 0.2$) pour tenir compte de ces approximations.

8.4.3 Algorithme de raffinement

L'algorithme de raffinement doit permettre de décider si un lien peut être établi entre un élément source et un élément récepteur donnés, au vu de la précision imposée par l'utilisateur. Si oui, il doit alors choisir le type de lien le mieux adapté au transfert. Sinon, il décide de raffiner l'émetteur, le récepteur, ou l'obstacle éventuel en se basant sur l'efficacité de chaque choix. Conformément à la stratégie de raffinement décrite à la section précédente, nous procédons comme suit (Voir également l'organigramme sur la figure 8.6) :

Disposant d'un élément émetteur S , d'un récepteur R et d'une limite ϵ sur l'erreur, nous estimons tout d'abord une borne supérieure sur le transfert d'énergie. Si trop peu

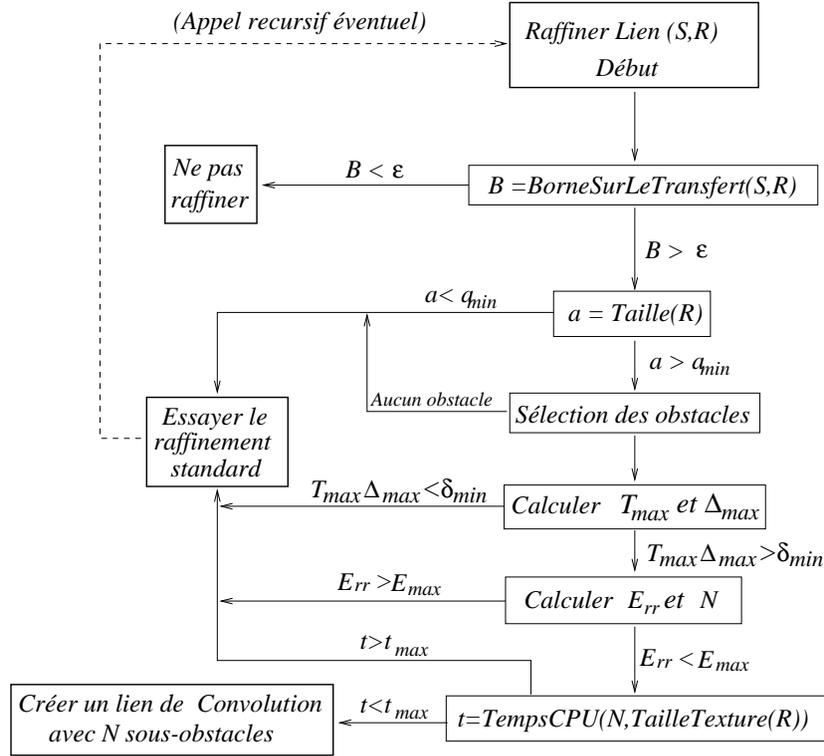


FIG. 8.6: Organigramme de l'algorithme de raffinement.

d'énergie est échangée entre les éléments, il n'est pas besoin de raffiner car l'erreur est déjà suffisamment faible. Dans ce cas, on ne raffine pas, ce qui veut dire qu'on crée un lien standard s'il n'en existe pas déjà un.

Sinon, on effectue successivement les tests décrits en 8.4.2, dans l'ordre le plus économique : nous comparons d'abord la taille du récepteur à la taille minimale absolue a_{min} . Nous sélectionnons ensuite les obstacles éventuels avant de calculer Δ_{max} et T_{max} .

Si l'une des trois conditions n'est pas satisfaite, l'algorithme essaye alors d'établir un lien standard, en raffinant éventuellement le récepteur ou l'émetteur, ce qui constitue un appel récursif à la procédure de raffinement.

Dans le cas contraire, la configuration permet l'établissement d'un lien de convolution. Il faut néanmoins s'assurer que ce lien permettra de calculer le transfert avec la précision requise à un coût raisonnable. Nous estimons pour cela le degré de subdivision de l'obstacle nécessaire à l'obtention d'une texture de visibilité dont l'erreur d'approximation est inférieure à un seuil ϵ_{max} fixé par l'utilisateur. Cette estimation est effectuée grâce à l'algorithme de raffinement décrit en 7.5 limité au raffinement de l'obstacle. Si la précision requise dans la texture de visibilité peut être atteinte, ce calcul renseigne sur le nombre N de sous-obstacles utilisés, *i.e* le nombre de convolutions et donc la quantité n de *FFT* qu'il faudra alors effectuer. Le temps de calcul pour une *FFT* sur une image de résolution $N \times N$ étant proportionnel à $N^2 \log(N)^2$, on a alors pour une texture de résolution $N \times N$:

$$n = 2 + 2N \quad \text{et donc} \quad t = (2 + 2N)N^2 \log(N)^2$$

Le nombre t représente le temps de calcul estimé pour le lien. Si t dépasse une valeur seuil t_{max} , le lien de convolution est abandonné.

Nous avons vu que dans plusieurs cas, le lien de convolution doit être abandonné pour essayer d'établir un lien standard entre l'émetteur et le récepteur. Un certain nombre de propriétés peuvent néanmoins être relevées au cours des calculs effectués pour effectuer ces tests :

Après le calcul des obstacles, on conserve tout d'abord la liste de ces obstacles pour accélérer leur sélection dans les appels récursifs éventuels de l'algorithme de raffinement.

L'information même d'existence de ces obstacles permet ensuite également d'accélérer l'établissement des liens standards, puisqu'on peut alors se passer d'effectuer des calculs de visibilité dans l'évaluation des facteurs de forme. La propriété de visibilité totale entre deux éléments, fournie de manière conservatrice par le *shaft-culling*, est de plus naturellement héritée lors de la subdivision et donc des appels récursifs.

Enfin, une fois calculé, le gradient de pénombre sert dans l'établissement des liens standard à calculer la taille minimale en deçà de laquelle il est inutile de raffiner le récepteur.

Il est important de noter que dans notre algorithme de raffinement, rien n'empêche explicitement l'établissement d'un lien de convolution à partir d'une surface ou d'un cluster quelconque, plutôt qu'une source lumineuse proprement dite. Dans la pratique, ce dernier cas est, à quelques exceptions près comme celui d'un puissant éclairage indirect, le seul ou des liens de convolutions sont effectivement établis, notamment à cause du critère de comparaison de la taille des polygones récepteurs et du gradient la fonction d'illumination.

8.4.4 Critère pour le raffinement standard

Lorsque les conditions nécessaires à l'établissement d'un lien de convolution entre un élément source et un élément récepteur ne sont pas satisfaites, nous avons vu que l'algorithme effectue un *raffinement standard*. La méthode de raffinement standard que nous utilisons est basée sur l'utilisation de bornes d'erreur *a posteriori*, et non *conservatives*, décrite à la section 2.3.3.

Les valeurs extrémales \bar{B}_j et \underline{B}_j de la radiosité sur l'émetteur intervenant dans le calcul sont estimées au moment du *Push/Pull* à partir des textures de radiosité et des valeurs uniformes propagées dans la hiérarchie selon un algorithme similaire à celui décrit par Lischinski [LSG94]. Les valeurs extrémales \bar{F}_{ij} et \underline{F}_{ij} des facteurs de forme sont estimées par échantillonnage en tirant parti des informations issues des tests d'établissement d'un lien de convolution.

Le seuil d'erreur pour le raffinement standard est noté ϵ_{st} , et A_{min} désigne l'aire minimale d'un récepteur pour un lien standard.

8.4.5 Paramètres du raffinement

En résumé, l'algorithme de raffinement est conduit par différents paramètres fixés par l'utilisateur, s'ajoutant aux paramètres du raffinement standard, et dont nous rappelons également la signification :

a_{min}	Taille minimale absolue pour un récepteur où aboutit un lien de convolution.
δ_{min}	Taille minimale relative du plus grand polygone d'un élément récepteur d'un lien de convolution.
E_{max}	Erreur maximale autorisée sur un lien de convolution.
t_{max}	Temps maximal autorisée pour le calcul du transfert d'énergie le long d'un lien de convolution.
ε	Seuil sur le transfert d'énergie indépendant de la visibilité, en dessous duquel on ne tente aucun raffinement.
A_{min}	Aire minimale d'un récepteur pour un lien standard.
ε_{st}	Seuil de contrôle de l'erreur pour un lien standard.

L'algorithme de raffinement est donc conduit par un grand nombre de paramètres. Dans le cadre d'une implémentation industrielle, il serait préférable de diminuer le nombre de ces paramètres. On pourrait par exemple corrélérer a_{min} et A_{min} . En revanche il est difficile d'attribuer à E_{max} et ε_{st} une valeur commune puisque les erreurs mesurées ne sont pas du tout de même nature.

Dans la section suivante, nous traitons de l'influence de ces différents paramètres sur le comportement du raffinement et la qualité du résultat.

8.4.6 Influence des paramètres de raffinement

Cette section regroupe un certain nombre de mesures expérimentales dont la synthèse permet de dégager la pertinence des paramètres contrôlant le raffinement. Certains paramètres n'ont de signification que par rapport aux dimensions de la scène (par exemple A_{min}). Pour ne pas dépendre de ces dimensions, nous re-dimensionnons toujours les scènes de manière à ce qu'elles rentrent au mieux dans une boîte de dimensions fixées. L'aire maximale d'un carré de support orthogonal à un des axes est donc invariable.

Répartition temporelle des liens de convolution

Le raffinement dépendant en partie de la répartition de la lumière dans la scène, elle-même calculée au fil des itérations, on est en droit de se demander quelle est la répartition temporelle de l'établissement de liens de convolution au fil des itérations. Nous avons relevé expérimentalement le comportement suivant, qui s'explique par ailleurs très facilement : les liens de convolution sont établis en quasi totalité lors de la première itération, sauf pour de grandes valeurs de ε . Dans ce cas la profondeur du raffinement n'atteint en effet pas toujours le niveau des sources de lumière dès la première itération, or les sources lumineuses sont des points de départ privilégiés des liens de convolution. Dans les expériences qui suivent, nous n'avons donc pas eu besoin de pousser très loin le nombre d'itérations pour effectuer des mesures valides concernant les liens de convolution.

Influence du type de lien sur le temps de calcul

Pour examiner l'influence du type de lien utilisé sur le temps de calcul, nous modifions E_{max} , tout en maintenant la borne inférieure du raffinement d'un lien standard A_{min} de façon à obtenir une précision comparable sur un récepteur maillé et un récepteur texturé.

Comme le montre la figure 8.7, le type de lien utilisé peut effectivement être réglé en modifiant E_{max} : une grande valeur de E_{max} autorise l'emploi de liens de convolution à haut niveau sans beaucoup raffiner l'obstacle, *i.e* à faible coût. Sur la figure 8.7 (à gauche) un unique lien de convolution suffit à effectuer le calcul de l'illumination sur le sol. En diminuant E_{max} , on empêche progressivement l'utilisation des liens de convolution jusqu'à un stade où plus aucun lien de convolution n'est utilisé. L'algorithme de raffinement est

alors obligé de limiter la taille des récepteurs de liens de convolution, et d'augmenter la profondeur du raffinement des obstacles (figure 8.7, *au centre*) pour satisfaire le critère de précision dans les textures de visibilité, dans un temps inférieur à t_{max} . Lorsque la précision requise ne peut plus être satisfaite (figure 8.7, *à droite*), les liens de convolution sont abandonnés au profit des liens standards.

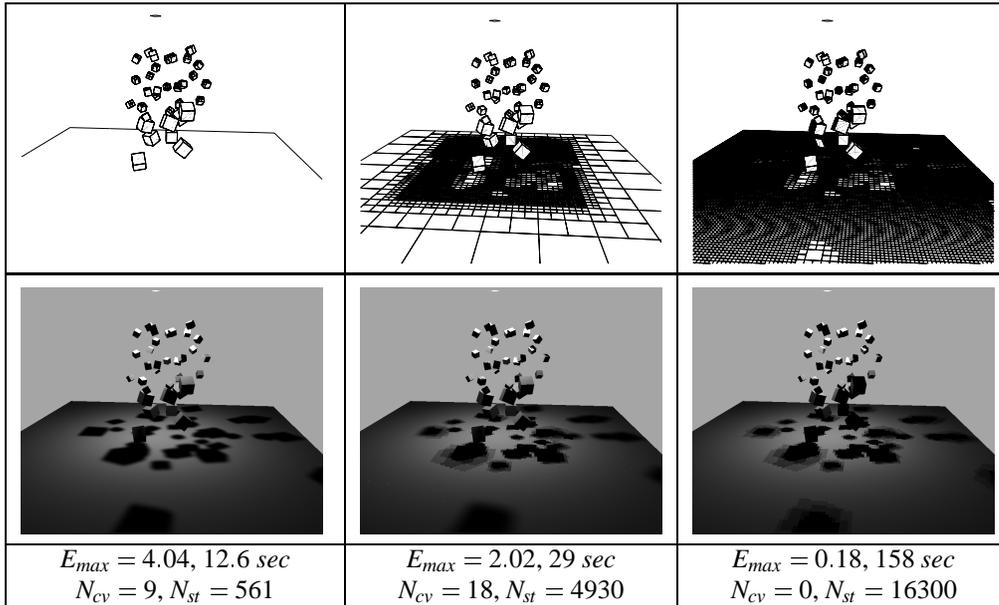


FIG. 8.7: Influence du type des lien utilisés sur le temps de calcul de la première itération. Le nombre N_{cv} de liens de convolution et N_{st} de liens standards correspond à l'ensemble des liens utilisés, mais la plupart (notamment à droite) aboutissent sur le sol. à gauche, des 9 liens de convolution indiqués, seul 1 aboutit au sol et illumine entièrement celui-ci. Le réglage de la proportion $\frac{N_{cv}}{N_{st}}$ s'obtient en agissant sur E_{max} (voir texte). Enfin, au centre, le maillage progressif entre la zone périphérique où aboutissent les liens de convolution et la zone centrale provient d'une contrainte sur la subdivision des surfaces.

Dans les trois images, nous avons utilisé

$$\epsilon_{st} = 0.034 \quad A_{min} = 0.079$$

Afin de laisser le maximum de liberté à l'algorithme de raffinement, nous avons attribué les valeurs suivantes aux autres paramètres :

$$\delta_{min} = 0 \quad t_{max} = 5.0 \text{ sec}$$

De cette expérience nous retiendrons ceci :

A précision égale, les temps de calculs d'une solution sont d'autant plus faibles qu'on favorise l'utilisation de liens de convolution.

Nous allons poursuivre cette étude expérimentale par une série de tests (figures 8.8, 8.9, 8.10, 8.11 et 8.12) conduits sur une scène permettant une grande latitude d'utilisation des liens de convolution, autant dans la taille des récepteurs que dans la subdivision des obstacles. Il s'agit de la scène représentée sur la figure 8.20, page 162.

Influence de ϵ_{st} sur le type des liens utilisés

Alors que E_{max} agit sensiblement sur la proportion des liens de convolution utilisés, on peut se demander ce qu'il en est de son homologue pour les liens standards ϵ . En effet, le raffinement étant un processus récursif, ϵ conditionne lui aussi, mais indirectement, le nombre de tentatives d'établissement de liens de convolution.

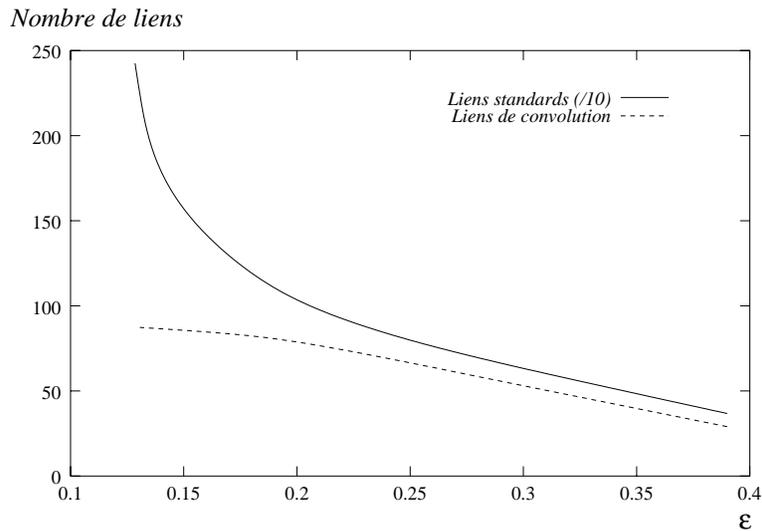


FIG. 8.8: Influence de E_{st} sur le type de lien utilisé.

Comme on le voit sur la figure 8.8, diminuer ϵ_{st} favorise la progression ascendante du nombre de liens de convolution et, dans une plus large mesure, du nombre de liens standards. Leur rapport reste à peu près constant sauf pour des faibles valeurs de ϵ_{st} où la proportion du nombre de liens de convolution s'affaiblit. Les liens standards sont en effet utilisés en grande partie dans des zones d'éclairage direct. Leur subdivision n'entraîne donc pas de nouveaux liens de convolution. Inversement, une fois établi, un lien de convolution n'est pas modifié par ϵ_{st} , ce qui explique que leur nombre ne peut diminuer.

Influence de t_{max} et E_{max} sur la subdivision des obstacles

L'expérience décrite par la figure 8.9 permet de vérifier que diminuer E_{max} force la subdivision des obstacles dans les liens de convolution. Dans le même temps, le nombre de liens de convolution diminue. Pour des valeurs suffisamment petites, la subdivision des obstacles ne permet en effet plus d'effectuer un calcul de transfert en un temps inférieur à t_{max} .

Le bilan en terme de nombre total de *FFT* n'est donc pas clairement représenté sur cette figure. En revanche, comme nous l'avons vu sur la figure 8.7, celui-ci finit toujours par diminuer.

Sur les figures 8.10 et 8.11, on peut voir que l'influence de t_{max} sur la profondeur (respectivement moyenne et maximale) de la subdivision des obstacles n'est pas uniforme : tant que t_{max} reste supérieur au temps requis par la subdivision des obstacles, imposée par la précision E_{max} , son influence est nulle. Ensuite, diminuer t_{max} limite naturellement la profondeur de la subdivision des obstacles.

Nous avons vu que la subdivision des obstacles, et donc le temps de calcul par lien, augmente en diminuant E_{max} . Par ailleurs, la taille moyenne des textures produites diminue. On ne sait donc pas *a priori* comment varie globalement le temps de calcul par lien. Les figures 8.10 et 8.11 nous montrent que la taille des textures utilisées prime sur la profondeur de la subdivision en terme de temps de calcul, puisque t_{max} limite prioritairement

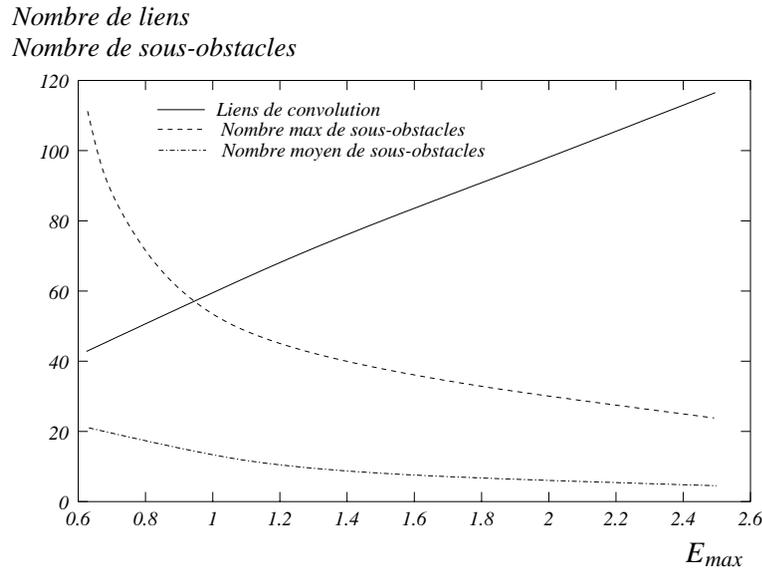


FIG. 8.9: Influence de E_{max} sur la nature des liens de convolution (voir texte)

la subdivision pour des grandes valeurs de E_{max} , *i.e* pour des tailles de texture élevées et un nombre de sous obstacles faibles, plutôt que le contraire.

Ce comportement se retrouve sur la figure 8.12 où nous avons représenté directement les tailles moyennes des textures en fonction de t_{max} pour différentes valeurs de E_{max} .

Conclusion

Nous avons montré au travers de ces diverses expériences la pertinence des paramètres de contrôle du raffinement. La durée des calculs et l’usage des liens de convolution peuvent en effet être contrôlés de façon relativement progressive en agissant sur ces différents paramètres.

8.5 Rendu

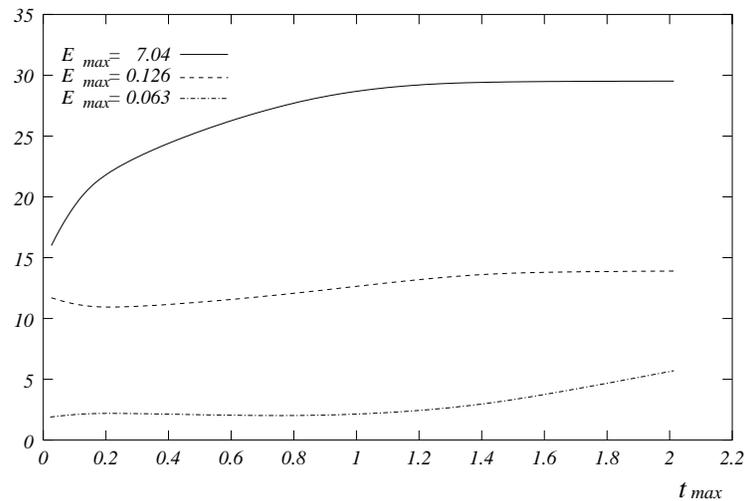
Nous expliquons dans cette section comment rendre la solution calculée, *i.e* comment mélanger de manière cohérente et en limitant les artefacts visuels la radiosit   issue des textures et des   l  ments hi  rarchiques uniformes.

8.5.1 Algorithme d’affichage

La radiosit   uniforme et les textures de radiosit   sont stock  es au niveau des feuilles de la hi  rarchie. Le rendu de la sc  ne est effectu   par un parcours en profondeur de la hi  rarchie et chaque polygone “feuille” est rendu. Pour afficher un polygone avec ses textures de radiosit   nous rendons tout d’abord le polygone avec sa radiosit   uniforme, puis nous ajoutons sa texture de radiosit   en le rendant une seconde fois, en ayant au pr  alable fix   les param  tres de *blending* de fa  on    ajouter les valeurs des pixels les unes aux autres. Au vu de l’algorithme utilis   pour le *Push/Pull*, la valeur de radiosit   uniforme d’un polygone comprend   ventuellement la moyenne de la radiosit   de ses textures, il faut donc la retrancher au moment du rendu. L’algorithme permettant d’effectuer le rendu des polygones est d  crit par le pseudo-code de la figure 8.13.

Comme nous l’avons expliqu   au chapitre pr  c  dent (Section 7.4.4.(c)) il faut prendre garde, lors du rendu,    forcer la librairie graphique    correctement moduler les textures,

Nombre de sous-obstacles

FIG. 8.10: Profondeur moyenne de subdivision des obstacles selon t_{max}

notamment lorsque les valeurs de couleur utilisées pour cette modulation n'appartiennent pas à l'intervalle $[0, 1]$. Nous appliquons pour cela la technique décrite en 7.4.4.(c).

8.5.2 Visibilité intra-récepteur

Par construction, la méthode de convolution utilisée pour le calcul du transfert d'énergie le long d'un lien de convolution ne prend pas en compte les phénomènes de visibilité à l'intérieur du récepteur. La visibilité intra-récepteur peut néanmoins être prise en compte à trois niveaux distincts dans notre algorithme :

Par subdivision

Comme pour toute méthode hiérarchique basée sur l'utilisation de *clusters*, cet apparent défaut peut être considéré comme une approximation naturelle due au niveau élevé d'établissement des liens. L'approximation peut en effet être réduite par subdivision du récepteur, considérant ainsi une partie de celui-ci comme obstacle potentiel pour la partie restante. C'est de loin la méthode la plus élégante et la plus naturelle de traiter le problème.

Dans le calcul du maillage d'affichage

Il est possible de prendre en compte l'ombre portée du récepteur sur lui-même au cours du calcul du maillage d'affichage en intégrant la visibilité dans le calcul de l'éclairage direct aux nœuds du maillage d'affichage, mais en limitant les obstacles considérés aux polygones du récepteurs, pour en optimiser le coût. Il s'agit d'une méthode génératrice d'artefacts visuels, puisque la visibilité n'est calculée qu'aux nœuds du maillage d'affichage. Par ailleurs, le rôle du maillage d'affichage est de contenir des valeurs d'éclairage direct, raison pour laquelle il est assez grossier. Il serait donc peu judicieux de l'utiliser pour stocker des informations de visibilité au prix d'une augmentation de sa résolution.

Au moment du rendu

La question de la visibilité intra-récepteur peut enfin être traitée au moment du rendu, où un certain nombre d'ombres portées visuellement importantes peuvent faire défaut. Bien que le fait de négliger ces ombres entre naturellement dans le cadre des approximations des

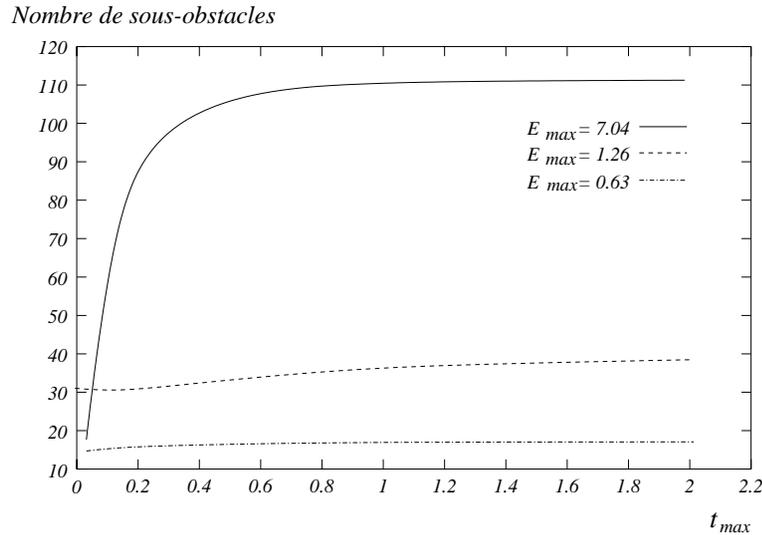


FIG. 8.11: Profondeur maximale de subdivision des obstacles selon t_{max}

méthodes hiérarchiques, il peut-être intéressant de les simuler au moment du rendu pour améliorer la qualité des images. Nous proposons pour cela d'utiliser l'extension *DepthTextureEXT* d'**OpenGL** dans un algorithme de rendu en trois passes (Voir également la figure 8.14) :

1. le récepteur est rendu *off-screen* depuis le centre de la source et la carte I_z des valeurs de profondeurs des pixels est lue dans l'image obtenue (Figure 8.14.a). On relève également la matrice M_1 de la projection utilisée.
2. le récepteur est ensuite rendu *off-screen*, depuis le point de vue de l'observateur en spécifiant que M_1 doit être utilisée pour transformer chaque pixel dessiné pour connaître sa profondeur vu du centre de la source. Il est alors possible de spécifier à **OpenGL** (Grâce à l'extension *DepthTextureEXT*) de ne dessiner que les pixels dont la profondeur ainsi calculée est inférieure à celle du pixel correspondant dans l'image I_z . Lorsque la profondeur d'un pixel est supérieure, celui-ci est considéré comme étant dans l'ombre de la source.
3. l'image ainsi calculée est utilisée pour construire un masque (Figure 8.14.b) au travers duquel est finalement dessiné le récepteur avec sa texture de radiosit . Les pixels passant au travers du masque sont ceux ne se trouvant pas dans l'ombre propre du r cepteur (Figure 8.14.c). Le masquage est r alis  automatiquement gr ce au *stencil buffer* d'**OpenGL**.

8.5.3 Le lissage de Gouraud

Comme nous l'avons vu en 1.2.5.(a), le lissage de Gouraud (ou *smooth-shading*) est une technique consistant   calculer, au moment de l'affichage, la couleur des pixels d'un polygone par interpolation   partir des couleurs aux sommets du polygone. Cette technique am liore la qualit  de l'image rendue en supprimant les discontinuit  d'ordre z ro de la radiosit .

Lorsque la radiosit  est repr sent e   la fois par des valeurs uniformes et par des textures, on ne peut pas calculer les valeurs aux n uds du maillage comme  tant une moyenne pond r e de la radiosit  des mailles adjacentes   chaque sommet. En effet, en ces points, une partie de la couleur est impos e par la contribution des textures de radiosit , qui correspond   la valeur du pixel de la texture.

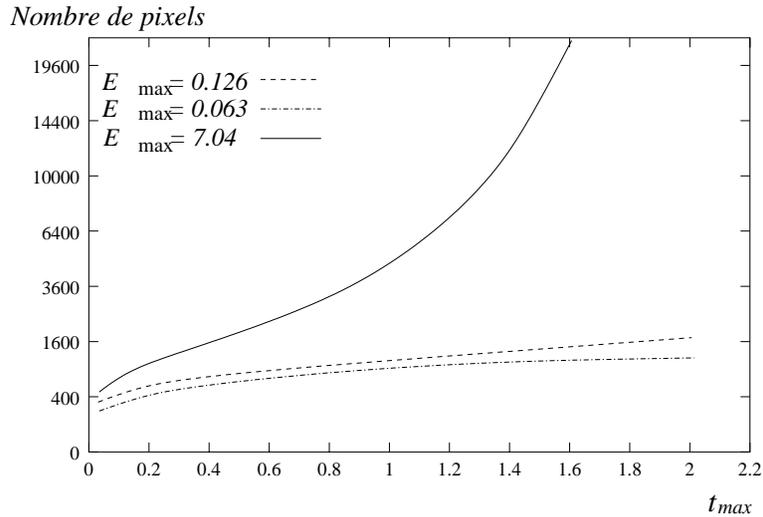


FIG. 8.12: Nombre moyen de pixels par texture en fonction de t_{max} (voir texte). L'échelle est logarithmique.

```

RenderPolygon(Polygone P)
  SetColor(P→Radiosity() - P→Textures()→MeanValue() )
  DrawSimplePoly(P)

  Enable(BLENDING)
  SetZBufferFunction(≤)
  ForAll Tin P.Textures()
    n = SmallestIntegerLargerThan(T→MaximumLuminance())
    for i = 1 to n
      DrawPolyWithTexture(P,T,n)
  Disable(BLENDING)

```

FIG. 8.13: Algorithme de rendu ajoutant la radiosité de textures à la radiosité uniforme.

Nous ne pouvons donc pas nous contenter d'ajouter les textures de radiosité au rendu interpolé des valeurs de radiosité uniformes, comme le montre la figure 8.15. Cette difficulté provient du fait que les valeurs calculées aux nœuds du maillage pour le *smooth-shading* sont globales et ne peuvent donc pas concorder avec les valeurs des textures de radiosité en ces points, qui elles, sont locales.

Pour obtenir une parfaite concordance de l'illumination des polygones partageant un sommet, nous décidons donc à l'avance d'une valeur de référence la plus juste possible, et nous forçons, au moment du rendu, les différents éléments de maillage partageant ce sommet, à concorder en ce point :

Au moment du *push/pull*, une valeur de radiosité uniforme v_I est attribuée à chaque sommet du maillage par moyennage des valeurs de radiosité uniforme des mailles voisines avec les poids décrits sur la figure 1.15. Cette valeur a un caractère global. Dans le même temps, une valeur de radiosité de textures de référence v_R est attribuée à chaque sommet du maillage. Cette valeur vaut 0 si aucune des mailles adjacentes n'admet de texture de radiosité. En revanche, Si l'un au moins des polygones adjacent admet de la radiosité de texture, la valeur choisie est le maximum des valeur des textures de radiosité des polygones

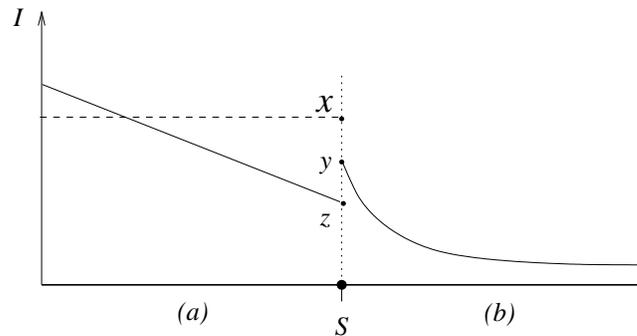
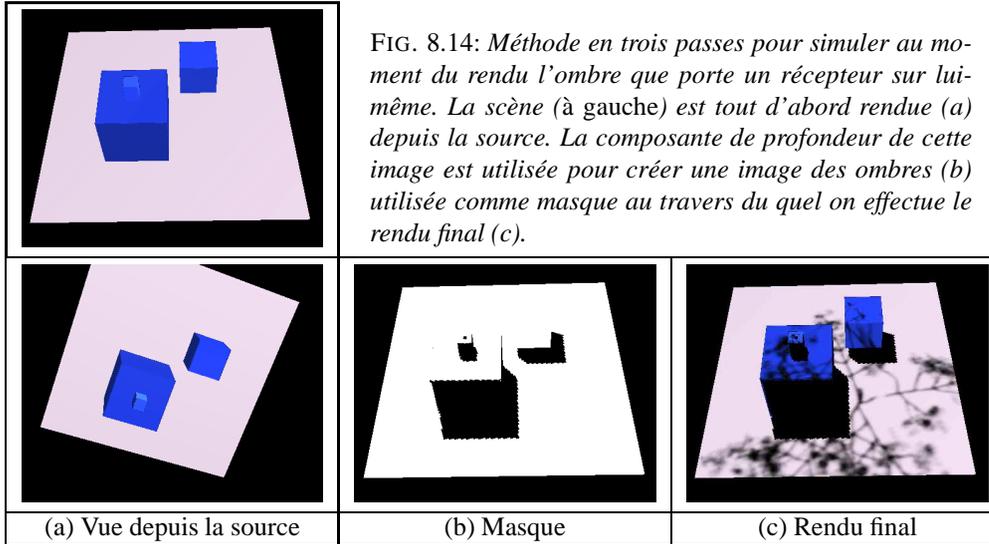


FIG. 8.15: Cette figure illustre la raison pour laquelle on ne peut pas rendre les textures de radiosit  par dessus la radiosit  uniforme apr s interpolation. La figure repr sente la radiosit  calcul e le long de deux ar tes (a) et (b) partageant un sommet S , l'ar te (a) appartenant   une maille A et l'ar te (b)   une maille B . On suppose de plus que A ne re oit que de la radiosit  uniforme (valeur x , courbe en pointill ) et B que des textures de radiosit  (courbe de droite, valeur y en S). La valeur z attribu e   S pour le smooth-shading n'a alors aucune raison de concorder avec y et une discontinuit  appar it.

adjacents   ce sommet. Cette valeur a un caract re local.

Ayant fix    chaque sommet une valeur de radiosit  uniforme et une valeur de radiosit  de texture de r f rence, le rendu est effectu  en rendant les textures par dessus la radiosit  interpol e   partir de la valeur calcul e en chaque sommet S comme :

$$v_S = v_I + v_R - T(S)$$

o  $T(S)$ est la valeur totale des textures de radiosit  au sommet S du polygone affich . L'utilisation du *maximum* dans le calcul de v_R permet d'assurer que v_S est toujours positive et donc jamais modifi e par **OpenGL** lors du rendu.

En proc dant ainsi, chaque polygone admettant le n ud S du maillage comme sommet est rendu, apr s accumulation de la radiosit  des textures et de la radiosit  uniforme, avec la m me valeur au point S : $v_I + v_R$. Dans l'exemple de la figure 8.15, on a donc $v_I = z$ et $v_R = y$. L'ar te (a) est donc rendue avec $v_S = z + y$ et l'ar te (b) avec $v_S = z + y - y = z$. Apr s ajout des textures, on a donc bien la m me valeur en v_S pour les deux ar tes.

Cette m thode est apparemment discutable puisqu'elle ne donne pas aux sommets la

valeur qui y a effectivement été calculée (parfois avec raison, notamment dans le cas où un des polygones adjacents n'admet que de la radiosit  de textures), mais l'erreur produite n'est que la r percussion de l'approximation d'une certaine proportion de la radiosit  par des valeurs uniformes.

Le *smooth-shading* ne permet cependant pas d' liminer certaines discontinuit s observables par exemple le long d'une ar te partag e par un polygone ayant une texture de radiosit  et un polygone n'en n'ayant pas. De telles discontinuit s sont dues   la diff rence de variation de la radiosit  sur les deux polygones ; l'une varie en effet approximativement comme $x \mapsto \frac{1}{1+x^2}$ et l'autre de fa on affine (C.f figure 8.16).

Pour supprimer de tels artefacts, il faut alors subdiviser le polygone n'ayant pas de texture de radiosit  et contraindre la subdivision pour qu'elle co ncide, le long de l'ar te commune aux deux polygones, au maillage d'affichage de la texture de radiosit  de l'autre polygone. Le probl me se complique encore lorsque les deux polygones ont des textures de radiosit  en nombre diff rent, avec des maillages d'affichage de r solutions diff rentes. La m thode pr c dente se g n ralise alors au cas de plusieurs textures apr s raffinement des maillages d'affichage pour mettre en accord leurs r solutions respectives.

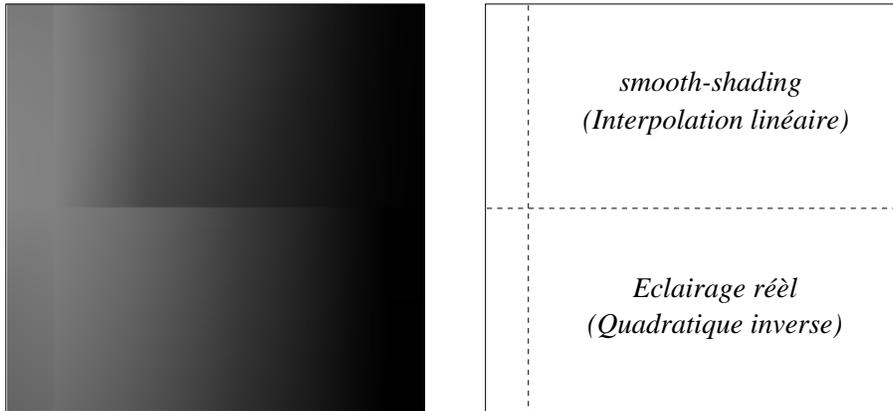


FIG. 8.16: Exemple de discontinuit  observ e lorsqu'un polygone textur  (en bas) jouxte un polygone n'ayant que de la radiosit  uniforme (en haut). Bien qu'elle permette la concordance des valeurs de radiosit  aux sommets communs des polygones, l'utilisation du *smooth-shading* produit alors une variation lin aire de la radiosit  qui contraste avec la variation en $\frac{1}{1+x^2}$ de l' clairage direct associ  au maillage d'affichage de la texture de radiosit , le long de l'ar te commune aux deux polygones.

8.5.4 Cas des environnements textur s

Dans le cas de sc nes poss dant des textures au sens classique du terme, *i.e* des images repr sentant de la r flectance, l'algorithme de rendu est modifi  de fa on   prendre en compte ces textures de mani re sp cifique. La r flectance n'est en effet plus incorpor e dans la composante d' clairage direct du maillage d'affichage mais prise en compte au moment du rendu, o , apr s avoir rendu le polygone avec sa valeur uniforme et ses textures de radiosit , nous multiplions point- -point l'image obtenue par celle de la texture de r flectance. Pour cela, nous rendons une nouvelle fois le polygone, mais  quip  de la texture de r flectance, sans modulation, et en fixant les param tres de *blending* de mani re   multiplier point- -point les pixels rendus et ceux d j  pr sents dans l'image. Le pseudo-code permettant de r aliser cette op ration est pr sent  sur la figure 8.17.

Lors du rendu de la texture de r flectance, il faut prendre garde   fixer la fonction de comparaison des valeurs de z de mani re   ne pas  carter les nouveaux pixels, car leur z

est exactement celui du pixel sur lesquels ils se trouvent. Nous utilisons donc pour cela une comparaison non stricte.

```

RenderTexturedPolygon(Polygon P)
  RenderPolygon(P)
  SetZBufferFunction(≤)
  Enable(GL_BLEND)
  glBlendFunc(GL_DST_COLOR, GL_ZERO)
  Disable(GL_BLEND)

```

FIG. 8.17: Algorithme de rendu ajoutant la radiosité de textures à la radiosité uniforme.

Un exemple de scène texturée est montré sur la figure 8.18. La figure représente la même scène, avec et sans les textures de réflectance. Dans le second cas, celles-ci sont remplacées par une valeur uniforme sur tout le polygone. Cet exemple permet de se rendre compte que la précision acquise dans les textures de radiosité est en partie masquée par l’affichage de textures de réflectance. Il nous paraît donc judicieux de prendre en compte la présence de textures de réflectances dans l’évaluation [FPSG96, MK94].



FIG. 8.18: Rendu des textures de radiosité dans une scène contenant des textures de réflectance. à gauche les textures de réflectance ont été enlevées pour mieux voir les textures de radiosité.

8.5.5 Optimisation du rendu

A résolution égale, l’affichage d’une texture est largement plus rapide que l’affichage d’un maillage fin. On peut donc avoir intérêt dans certains cas à privilégier l’utilisation des textures de radiosité. Cet effet est obtenu en relaxant certains paramètres de l’algorithme de raffinement.

En revanche, comme nous le verrons à la section 8.6, la mémoire de textures est un facteur limitant de notre algorithme. Selon les machines, lorsque la mémoire de texture est séparée de la mémoire centrale, il peut donc être utile de limiter le nombre de textures affichées. Nous proposons pour cela deux méthodes :

Il est tout d’abord possible de convertir un polygone ayant une ou plusieurs textures de radiosité en maillage d’éléments à radiosité uniforme. Il suffit pour cela de subdiviser l’élément et d’effectuer une *Push/Pull* pour mettre à jour la radiosité dans la hiérarchie créée. Au cours de ce *Push/Pull*, les textures de radiosité descendues au niveau des feuilles sont éliminées après en avoir conservé la valeur moyenne qui est alors ajoutée à la radiosité

uniforme de chaque feuille. Cette méthode allège la mémoire de textures et encombre la mémoire centrale.

Une autre manière d'économiser de la mémoire de texture est de *mixer* les différentes textures de radiosité attachées à un même polygone. Pour cela, nous rendons le polygone avec ses différentes textures de radiosité et l'image obtenue est utilisée comme nouvelle texture de radiosité. A la différence de la méthode précédente, cette optimisation condamne le raffinement de la solution, car les différentes textures de radiosité ne sont pas nécessairement produites par la même source. Il n'est donc plus possible de calculer les valeurs d'éclairage direct du maillage d'affichage correspondant si l'on subdivise les polygones sur lesquels se trouve la texture issue de leur combinaison. Cette méthode permet cependant d'accélérer le rendu de la scène et de diminuer la taille des données représentant la solution.

8.6 Résultats

Nous présentons dans cette section des simulations de l'éclairage effectuées grâce à notre méthode.

La figure 8.19 présente une comparaison de notre méthode avec une méthode de radiosité hiérarchique classique utilisant un critère de raffinement de type *BF* [HSA91]. La solution calculée avec des liens de convolution a pleinement convergé en 20.2sec. Afin d'effectuer la comparaison, nous avons réglé empiriquement les paramètres de cette dernière pour qu'elle converge dans le même temps. Comme on peut le constater sur les images présentées, les liens de convolution ont rapidement saisi des détails d'ombre relativement fins (Par exemple les quatre ombres de la lampe de bureau sur le mur). Le nombre de liens de convolution pour ce calcul est de dix-huit : quatre pour le sol et deux des murs, les autres arrivant sur les bureaux. Comme aucun objet ne projette d'ombre dessus, le mur du fond (*en haut à droite* sur la figure 8.22) n'a reçu que des liens standards, et se trouve régulièrement subdivisé. Dans cette simulation, le raffinement du maillage a donc été adapté au calcul de l'équilibre des échanges lumineux indépendamment de la visibilité.

Dans le temps imparti, la méthode de radiosité hiérarchique (*en bas*) a correctement simulé la répartition globale de l'énergie lumineuse mais n'a que grossièrement calculé la position des ombres.

La figure 8.20 prouve que notre méthode permet d'effectuer des simulations dans des environnements où la visibilité est très complexe. La solution présentée sur cette figure a été calculée en 45 secondes sur une *Onyx*². La scène contient 13000 polygones en entrée dont deux sources de lumière étendues. Le nombre de liens de convolution utilisés est de 48.

La figure 8.21 montre le maillage utilisé pour la solution de la figure 8.20, et confirme que celui-ci n'est plus conduit que par le critère de précision sur les échanges lumineux indépendamment de la visibilité.

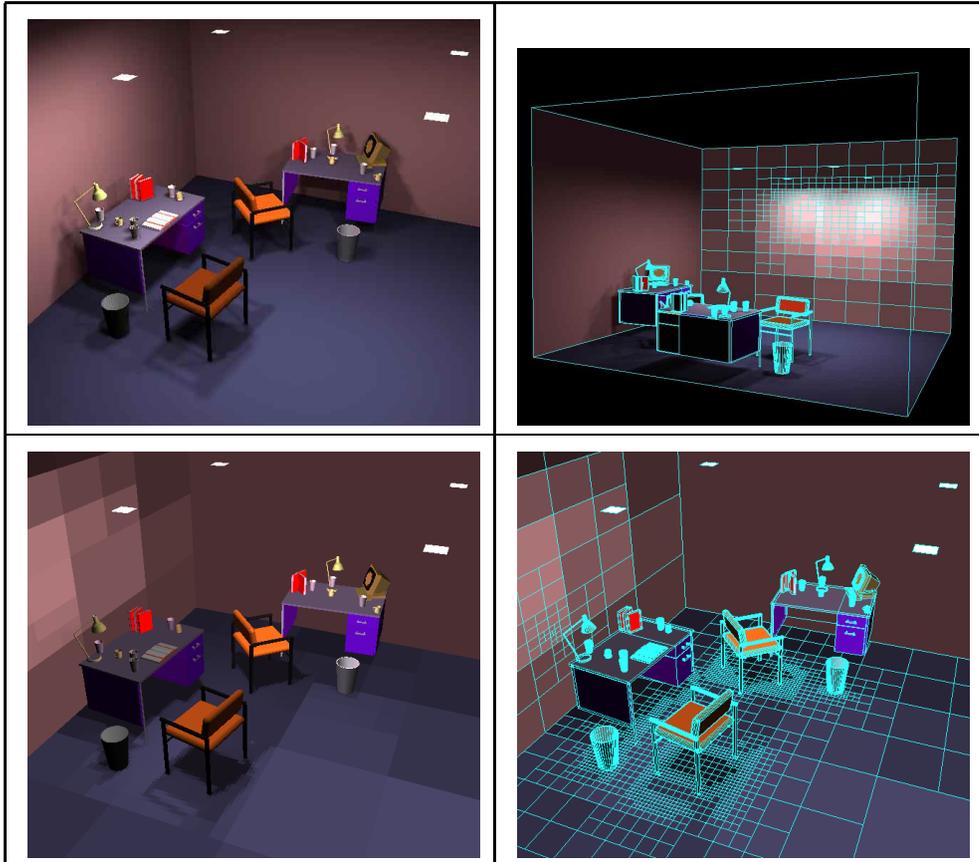


FIG. 8.19: *Comparaison avec la méthode de radiosité hiérarchique classique. La solution en haut, à gauche et à droite a été calculée par notre méthode en 20.2sec sur une Onyx². Pour comparaison, la solution présentée en bas, à gauche et à droite a été calculée dans le même temps par une méthode de radiosité hiérarchique classique. La scène contient 5380 surfaces en entrée.*

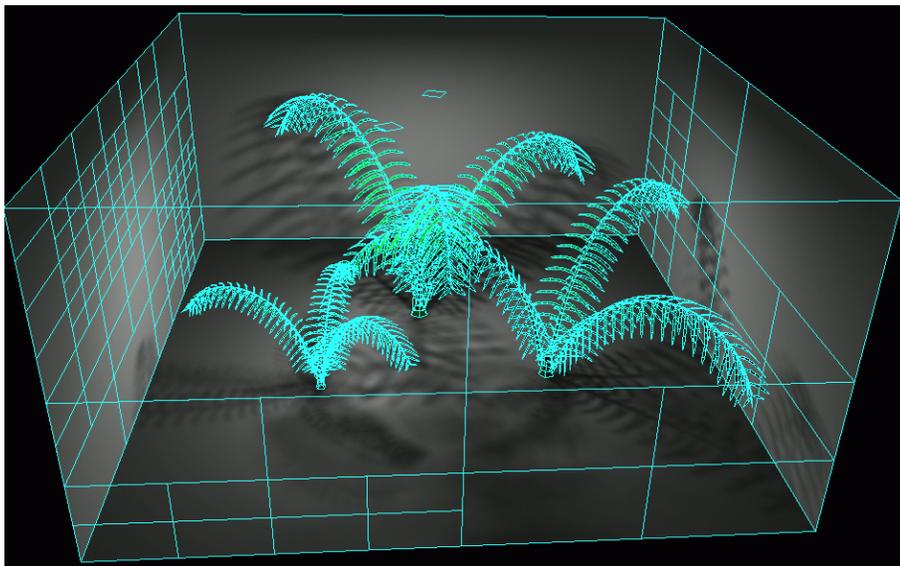


FIG. 8.21: *Maillage de la solution présentée sur la figure 8.20.*

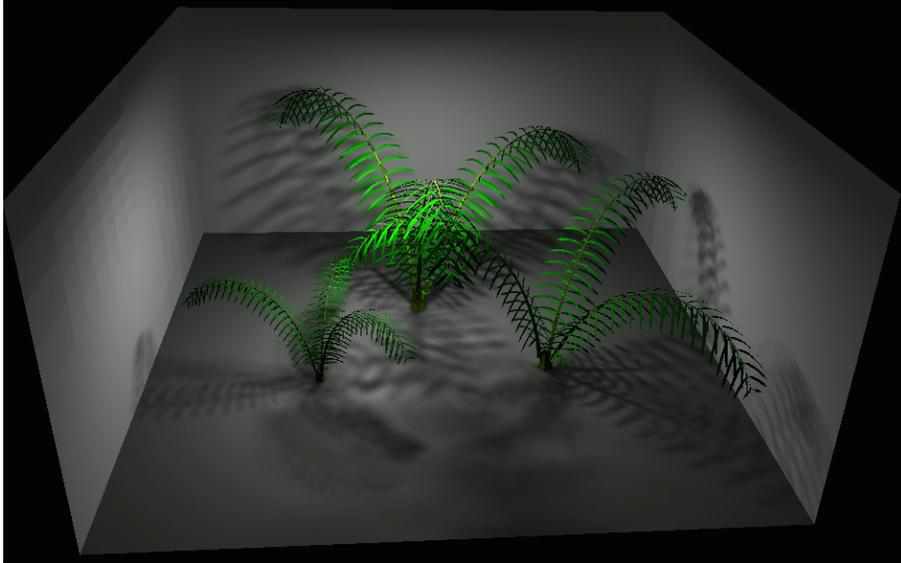


FIG. 8.20: Exemple de scène à visibilité complexe (13000 polygones, 45sec sur une Onyx²).

Sur la figure 8.22, on peut observer le détail du mur de droite de la figure précédente. Afin de vérifier que le mélange des ombres calculées par convolution et celles obtenues par raffinement du maillage se superposent de manière cohérente, nous avons manuellement forcé le raffinement d'un des liens de convolution arrivant sur le quart inférieur gauche du mur tout en interdisant l'usage de nouveaux liens de convolution. L'ombre projetée par l'une des deux source a donc de fait été simulée par raffinement du maillage. Comme on peut le voir sur la figure 8.22, les ombres se correspondent parfaitement. Le raffinement progressif du maillage dans les trois autres quart du mur est la conséquence d'une contrainte pour le *smooth-shading* (voir la section 1.2.5.(a)).

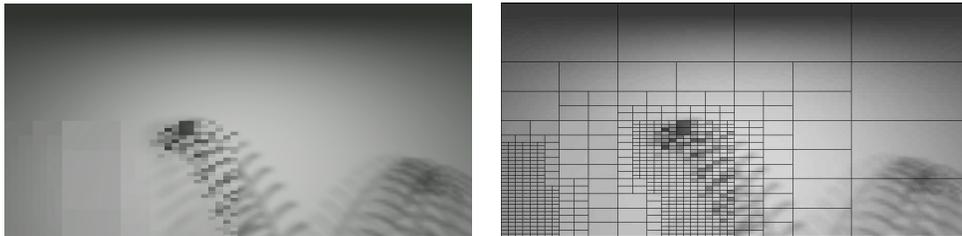


FIG. 8.22: Exemple de mélange de maillage et de textures. Un des deux liens de convolution arrivant sur le quart en bas à gauche a été artificiellement raffiné en liens standards produisant ainsi des éléments à radiosité uniforme. Les ombres se juxtaposent parfaitement avec celles des quarts voisins ainsi que celle de l'autre lien de convolution arrivant sur le même quart.

8.7 Conclusion

Nous avons présenté dans ce chapitre une méthode de hiérarchique de radiosité permettant d'effectuer des simulations de haute qualité visuelle dans des environnements complexes. Dans cette méthode, les calculs de la visibilité et de l'équilibre global de l'énergie lumineuse sont dé-corrélés et bénéficient chacun d'un traitement approprié : en plus des

liens apparaissant habituellement en radiosité hiérarchique, qui propagent des valeurs spectrales de radiosité, des liens pour lesquels le transfert d'énergie est calculé par la méthode de convolution présentée au chapitre 7 sont utilisés dans les situations complexes de visibilité. Avec les méthodes de radiosité hiérarchiques standards, de telles situations entraînent habituellement le raffinement du maillage dans un domaine où il n'est plus efficace. La méthode de convolution tire, elle, parti au moment du calcul de la cohérence de la visibilité sur tout le domaine d'action du lien, ce que ne fait pas le calcul séparé de nombreux liens classiques. Outre son facteur d'accélération appréciable, cette nouvelle méthode a l'avantage de réduire considérablement les artefacts visuels, puisque la méthode de convolution utilisée simule naturellement les variations progressives de pénombre.

Grâce à l'intégration cohérente du contrôle de la précision sur le calcul du transfert d'énergie le long d'un lien de convolution dans l'algorithme de raffinement, notre méthode permet d'échanger du temps de calcul contre de la précision. Cette capacité est à la base des algorithmes modernes de simulation puisqu'elle permet à l'utilisateur de positionner continuellement son calcul entre une solution rapide mais approximative permettant de mettre au point différents paramètres, et une solution extrêmement précise mais plus longue à obtenir, dans un contexte plus final.

Conclusion

Dans ce document, nous avons tenté d'apporter des réponses au problème du contrôle de l'erreur dans les méthodes hiérarchiques de radiosité. Du fait que les calculs de visibilité constituent la majeure partie du coût de ce type d'algorithmes, nous nous sommes plus particulièrement intéressés aux approximations de la visibilité. En vue de produire des méthodes offrant un positionnement continu entre des solutions très approximatives mais rapides et des simulations arbitrairement précises mais longues à obtenir, nous avons formulé ces approximations à l'aide de représentations hiérarchiques de la visibilité à travers les obstacles.

Grâce à une approche basée sur l'étude de la perturbation du système linéaire intervenant dans les méthodes hiérarchiques de radiosité, nous avons tout d'abord étudié l'impact des approximations des transferts d'énergie élémentaires issues de la discrétisation de l'équation de radiosité, sur l'erreur globale dans la solution du système linéaire. Deux approches différentes ont été décrites : une approche *a priori* que nous avons développée, et une approche *a posteriori*, issue de la littérature. En établissant clairement un lien bilatéral entre l'erreur locale et l'erreur globale, nous avons motivé, à travers ces deux approches, la confection d'algorithmes de calcul contrôlé des facteurs de forme. Grâce à cette étude préalable, nous savons en effet que de tels outils, en permettant d'assurer une précision donnée sur le calcul des transferts locaux, ou de façon duale, de fournir des bornes sur l'approximation obtenue dans un temps fixé, offrent la possibilité de contrôler par avance la précision de la simulation.

Nous avons alors présenté à la partie II le principe de tels algorithmes, basés sur le pré-calcul et le stockage d'informations multi-échelles de visibilité dans la hiérarchie des obstacles. Ces informations sont ensuite utilisées par l'algorithme pour générer des approximations contrôlées des facteurs de forme où interviennent ces obstacles. Nous avons ensuite discuté de l'implémentation de ces algorithmes dans le cadre du calcul de la radiosité en dimension deux et trois. Alors que la faible complexité géométrique propre à la deuxième dimension nous a effectivement permis d'appliquer efficacement ce type d'algorithme, leur implémentation en dimension trois, sans être impossible, est rendue inintéressante par le coût mémoire qu'ils représentent en comparaison de la qualité médiocre des bornes obtenues.

Le fait que l'œil humain soit sensible à une forme d'erreur difficilement caractérisable par des distances mathématiques simples, a d'autre part motivé des approches plus perceptives du contrôle de l'erreur due aux approximations de visibilité. Dans ce cadre, nous avons tout d'abord étudié au chapitre 6 la possibilité de reproduire artificiellement des ombres complexes à partir d'échantillons pré-calculés. Aucune des deux méthodes que nous avons étudiées n'a cependant permis de capturer la cohérence entre les différentes bandes de fréquence, qui constitue en quelque sorte la signature d'une image, sans produire des artefacts et autres discontinuités.

Plus proche du modèle géométrique, mais toujours basée sur une estimation perceptive de l'erreur, la méthode de convolution, que nous avons présentée au chapitre 7, constitue une élégante approximation multi-niveaux de la visibilité, et nous a permis de calculer

rapidement des ombres intrinsèquement exemptes des artefacts auxquels nous sommes naturellement sensibles.

Comme synthèse de nos travaux, nous avons finalement présenté en dernière partie, une méthode dans laquelle cohabitent différents algorithmes de contrôle de l'erreur, mais utilisés dans les conditions adéquates : un critère basé sur une distance mathématique pour contrôler l'équilibre global de l'énergie associé de manière cohérente à un critère perceptif pour le calcul des ombres.

Le travail que nous avons présenté ne correspond qu'à une partie de l'étude nécessaire à l'obtention d'un algorithme complet de contrôle de l'erreur pour le calcul de la radiosité.

Le pas suivant à franchir est tout d'abord l'implémentation d'une méthode de simulation de la radiosité incorporant dans un même système à la fois l'encadrement *a priori* de l'erreur développé dans la première partie de ce document, et les techniques de calcul contrôlé du facteur de forme décrites dans la partie II. Une telle méthode trouverait son application dans le cadre de simulations quantitatives des échanges radiatifs, dans des domaines où la notion de qualité d'une simulation est purement mathématique : botanique, séchage infra-rouge, calcul de solutions de référence pour la simulation de l'éclairage.

Lorsque la précision requise devient très élevée, un tel système ne peut plus se permettre de négliger les différentes autres approximations dont nous avons fait l'inventaire au chapitre 2. En amont doivent alors se greffer des méthodes d'estimation et de prise en compte de ces approximations.

Il n'est par exemple pas envisageable de pousser le raffinement du calcul des facteurs de forme au delà d'un point où l'erreur associée est du même ordre de grandeur que l'erreur due à la discrétisation de l'environnement. Au chapitre 2, nous avons brièvement étudié l'influence de la modification du maillage, et plus généralement de la hiérarchie des liens, sur nos critères d'estimation de l'impact des approximations de visibilité.

Il existe donc un lien intime entre ces deux biais de réduction de l'erreur. D'une manière générale, le rapport des critères d'erreur de discrétisation et d'approximation des transferts donne la direction optimale de l'effort de raffinement : maillage ou visibilité. En d'autres termes, les bornes d'erreur de discrétisation permettent de limiter le raffinement de la visibilité dans le calcul des facteurs de forme à un intervalle utile et inversement. Afin de comparer l'ordre de grandeur des deux sources d'erreur, nous préférons une estimation *a posteriori* plutôt qu'*a priori* de l'erreur due à l'approximation des transferts locaux, qui, basée sur des bornes d'erreur exactes en donne, certes avec une itération de retard, une estimation beaucoup moins pessimiste.

Dans les méthodes classiques de contrôle de l'erreur en radiosité, le contrôle de l'erreur s'effectue généralement par action sur le raffinement du maillage. Comme une estimation de cette erreur est obtenue par propagation de bornes *a posteriori* sur la radiosité, de telles méthodes mesurent en réalité l'impact des approximations des facteurs de forme. En raffinant le maillage, on effectue alors de façon déguisée un échantillonnage plus précis du calcul de ces facteurs de forme et donc de la visibilité.

D'autre part, en synthèse d'images, lorsque la qualité visuelle du résultat est un critère d'importance, c'est vers des méthodes du type de celle présentée à la partie IV que nous devons nous tourner. Dans ce cas, et dans l'optique d'une utilisation par des personnes étrangères au domaine de la simulation de l'éclairage, il est primordial de donner à l'utilisateur la possibilité d'agir de façon plus intuitive sur la qualité de la solution, idéalement sous la forme d'un unique curseur portant la mention *qualité*. Une telle démarche demanderait tout d'abord d'établir une relation entre les différents critères d'erreur (ombres et balance globale de l'énergie) qui restent dans notre implémentation actuelle totalement indépendants, la difficulté étant de comparer des notions de qualité perceptives et quantitatives.

L'erreur véhiculée par l'algorithme ne peut alors être rattachée à une quelconque signification physique. Pour ce faire, il serait bien sur possible de borner l'erreur le long

des liens de convolution, mais nous perdrons les avantages d'un critère perceptif. La seule chose que nous pouvons demander à une telle méthode, est que lorsque nous diminuons les seuils d'erreurs, le résultat du calcul converge vers une solution exacte, en un temps de plus en plus long. L'algorithme que nous avons présenté au chapitre 8 remplit ce contrat, puisqu'en réduisant les tolérances d'erreur, les liens de convolution sont progressivement abandonnés, au profit de liens classiques. Au cours de cette progression, les temps de calcul augmentent régulièrement de même que la qualité, sans que l'on puisse mathématiquement la quantifier.

Annexe A

Quelques résultats concernant les harmoniques sphériques

Les harmoniques sphériques sont des fonctions à deux variables construites à partir des *polynômes de Legendre associés*. Elles forment une base orthonormée de fonctions directionnelles et se prêtent donc naturellement à la représentation de telles fonctions. Les propriétés des harmoniques sphériques furent connues bien avant l'existence des ordinateurs. Elles sont énoncées pour la plupart dans l'ouvrage de référence de Hobson [Hob31].

Cette annexe rappelle la définition et les propriétés élémentaires des harmoniques sphériques, et présente quelques algorithmes et résultats très utiles dans leur maniement : évaluation efficace d'une combinaison linéaire d'harmoniques sphériques (section A.2), formule de rotation des harmoniques sphériques (section A.3). Bien qu'à première vue sans relation avec le travail que nous avons présenté jusqu'ici, ces résultats nous ont servi pour la représentation des diverses distributions directionnelles comme la *transmittance directionnelle* des obstacles en dimension trois, dans les expériences que nous avons menées. D'autre part, la formule de rotation des harmoniques sphériques que nous présentons à la section A.3 ainsi que son implémentation sont très peu connues en l'informatique graphique, bien que très utiles en pratique. Nous en présenterons des applications à la section A.3.4.

Les notations utilisées sont présentées sur la figure A.1. Les démonstrations de toutes les propriétés énoncées sont accessibles dans un rapport technique [Sol96].

P_l	Polynôme de Legendre d'ordre l
P_l^m	Polynôme de Legendre associé d'ordre l et m
Y_l^m	Harmonique sphérique complexe ($-l \leq m \leq l$)
C_l^m	Coefficient de décomposition sur les Y_l^m
y_l^m	Harmonique sphérique réelle ($-l \leq m \leq l$)
$c_{l,m}$	Coefficient de décomposition sur les y_l^m
$S(x)$	1 Si $x \geq 0$, 0 Sinon

FIG. A.1: Notations relatives aux harmoniques sphériques

A.1 Définitions et propriétés

Nous définissons tout d'abord les *polynômes de Legendre*, qui nous serviront essentiellement à introduire les *polynômes de Legendre associés* qui interviennent dans la formulation des harmoniques sphériques.

A.1.1 Polynômes de Legendre

Les *polynômes de Legendre* sont les solutions polynomiales P_n de l'équation différentielle :

$$\frac{d}{dx} \left[(1-x^2) \frac{dy}{dx} \right] + n(n+1)y = 0 \quad n \in \mathbb{N}$$

Ils s'écrivent :

$$P_n = \frac{1}{2^n n!} \frac{d^n}{dX^n} (X^2 - 1)^n$$

et forment une famille orthogonale pour le produit scalaire suivant :

$$\langle f, g \rangle = \int_{-1}^1 f(t)g(t)dt$$

Propriétés

Les polynômes de Legendre possèdent un nombre impressionnant de propriétés intéressantes [Hob31]. Les deux propriétés suivantes sont notamment essentielles :

– pour tout entiers n, n' :

$$\int_{-1}^1 P_n(x)P_{n'}(x)dx = \delta_{n,n'} \frac{2}{2n+1}$$

– la relation de récurrence suivante est également utile :

$$nP_n - (2n-1)XP_{n-1} + (n-2)P_{n-2} = 0$$

A.1.2 Polynômes de Legendre associés

Les *polynômes de Legendre associés* sont des fonctions définies sur $[0, 1]$ à partir des polynômes de Legendre par la relation suivante :

$$P_l^m = (-1)^m (1-X^2)^{\frac{m}{2}} \frac{d^m P_l}{dX^m}$$

Nous avons donc en particulier $P_l^0 = P_l$.

Propriétés

Les relations de récurrence suivantes permettent d'évaluer ces fonctions de façon efficace :

$$P_0^0 = 1$$

$$P_m^m = (1-2m)\sqrt{1-X^2}P_{m-1}^{m-1} \quad (\text{A.1})$$

$$P_{m+1}^m = (2m+1)XP_m^m \quad (\text{A.2})$$

$$P_l^m = \frac{2l-1}{l-m}XP_{l-1}^m - \frac{l+m-1}{l-m}P_{l-2}^m \quad l \geq m+2 \quad (\text{A.3})$$

Nous avons de plus :

$$P_l^{m+2} + 2(m+1)\frac{X}{\sqrt{X^2-1}}P_l^{m+1} - (n-m)(n+m+1)P_l^m = 0 \quad l \geq m$$

A.1.3 Le théorème d'addition

Ce théorème donne accès à la formule de rotation des harmoniques sphériques (Voire section A.3) dans un cas particulier. Il s'énonce ainsi :

Théorème 3 ([Hob31] p418) Pour tout $(\theta, \theta_0) \in [0, \pi]^2$, $\varphi \in [0, 2\pi]$, nous avons :

$$P_n(\cos \theta \cos \theta_0 + \sin \theta \sin \theta_0 \cos \varphi) = P_n(\cos \theta)P_n(\cos \theta_0) + 2 \sum_{m=1}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta)P_n^m(\cos \theta_0) \cos(m\varphi) \quad (\text{A.4})$$

On peut déduire de ce théorème une formule qui nous servira par la suite. Multiplions chaque membre de (A.4) par $\cos \varphi$ avant d'intégrer sur $[0, 2\pi]$:

$$\int_0^{2\pi} \cos(m\varphi) P_n(\cos \theta \cos \theta_0 + \sin \theta \sin \theta_0 \cos \varphi) d\varphi = 2\pi \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta)P_n^m(\cos \theta_0) \quad (\text{A.5})$$

A.1.4 Harmoniques sphériques

On note S^3 la sphère unité de \mathbb{R}^3 centrée en 0. Elle représente également l'ensemble des vecteurs unitaires de \mathbb{R}^3 , ou encore l'ensemble des directions issues d'un point. Tout point de S^3 est repéré de manière unique par deux angles θ et φ , comme le suggère la figure A.2.

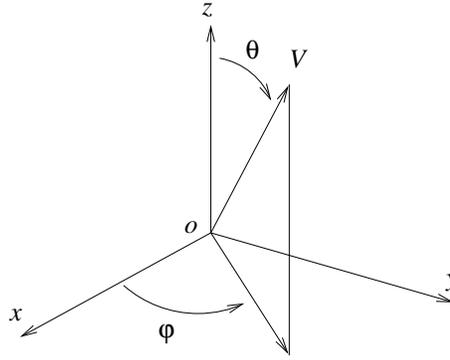


FIG. A.2: Repérage d'une direction par deux angles

Ses coordonnées cartésiennes dans \mathbb{R}^3 s'écrivent alors :

$$\begin{cases} x = \sin \theta \cos \varphi \\ y = \sin \theta \sin \varphi \\ z = \cos \theta \end{cases}$$

Les harmoniques sphériques sont alors définies en fonction de θ et φ par :

$$Y_l^m(\theta, \varphi) = \frac{N_l^{|m|}}{\sqrt{2}} e^{im\varphi} P_l^{|m|}(\cos \theta) \quad l \in \mathbb{N}, \quad -l \leq m \leq l$$

avec :

$$N_l^m = \sqrt{\frac{2l+1}{2\pi} \frac{(l-|m|)!}{(l+|m|)!}}$$

Le nombre l est appelé *degré* de l'harmonique Y_l^m . Les fonctions Y_l^m forment une famille orthonormée pour le produit scalaire

$$(f, g) = \int_{S^3} fg = \int_{\theta=0}^{\pi} \int_{\varphi=0}^{2\pi} f(\theta, \varphi)g(\theta, \varphi) \sin \theta d\theta d\varphi$$

Pour une distribution directionnelle f , sa meilleure approximation \tilde{f} par des harmoniques sphériques de degré au plus L , au sens du produit scalaire (\cdot, \cdot) est donc définie par :

$$\tilde{f} = \sum_{l=0}^L \sum_{m=-l}^l C_l^m Y_l^m \quad \text{avec} \quad C_l^m = (Y_l^m, f)$$

Notons que les harmoniques Y_l^m et coefficients C_l^m vérifient toujours :

$$C_l^{-m} = \overline{C_l^m} \quad \text{et} \quad Y_l^{-m} = \overline{Y_l^m} \quad (\text{A.6})$$

A.1.5 Harmoniques sphériques pour une fonction réelle

Pour une fonction f à valeurs dans \mathbb{R}^3 , un certain nombre de simplifications apparaissent et permettent une représentation par des *harmoniques réelles*. Soit $f : \mathcal{S}^3 \rightarrow \mathbb{R}$. Sa décomposition en harmoniques sphériques s'écrit :

$$f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l C_l^m Y_l^m(\theta, \varphi)$$

i.e :

$$f = \sum_{l=0}^{\infty} \left[C_l^0 Y_l^0 + \sum_{m=1}^l (C_l^m Y_l^m + C_l^{-m} Y_l^{-m}) \right]$$

donc d'après (A.6) :

$$\begin{aligned} f &= \sum_{l=0}^{\infty} \left[C_l^0 Y_l^0 + \sum_{m=1}^l (C_l^m Y_l^m + \overline{C_l^m} \overline{Y_l^m}) \right] \\ &= \sum_{l=0}^{\infty} \left[C_l^0 Y_l^0 + \sum_{m=1}^l (Re(C_l^m) [Y_l^m + \overline{Y_l^m}] + iIm(C_l^m) [Y_l^m - \overline{Y_l^m}]) \right] \\ &= \sum_{l=0}^{\infty} \left[C_l^0 Y_l^0 + \sum_{m=1}^l (2Re(C_l^m)Re(Y_l^m) - 2Im(C_l^m)Im(Y_l^m)) \right] \\ &= \sum_{l=0}^{\infty} \left[C_l^0 Y_l^0 + \sum_{m=1}^l 2Re(C_l^m)Re(Y_l^m) + \sum_{m=-l}^{-1} -2Im(C_l^{-m})Im(Y_l^{-m}) \right] \end{aligned}$$

On pose donc :

$$c_{l,m} = \begin{cases} \sqrt{2}Re(C_l^m) & \text{si } m > 0 \\ C_l^0 & \text{si } m = 0 \\ -\sqrt{2}Im(C_l^{-m}) & \text{si } m < 0 \end{cases} \quad (\text{A.7})$$

et :

$$y_l^m = \begin{cases} \sqrt{2}Re(Y_l^m) & \text{si } m > 0 \\ Y_l^0 & \text{si } m = 0 \\ \sqrt{2}Im(Y_l^{-m}) & \text{si } m < 0 \end{cases}$$

Nous avons alors :

$$f = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_{l,m} y_l^m$$

Nous avons donc :

$$\begin{cases} \text{si } m < 0 & c_{l,m} = -\sqrt{2} \text{Im}(c_l^{-m}) = \int_{S^3} -\sqrt{2} f \text{Im}(\overline{Y_l^{-m}}) \\ \text{si } m = 0 & c_l^0 = C_l^0 = \int_{S^3} f \overline{Y_l^0} = \int_{S^3} f Y_l^0 \\ \text{si } m > 0 & c_{l,m} = \sqrt{2} \text{Re}(C_l^m) = \int_{S^3} \sqrt{2} f \text{Re}(\overline{Y_l^m}) \end{cases}$$

Nous avons donc, dans tous les cas :

$$c_{l,m} = \int_{S^3} f y_l^m$$

avec

$$y_l^m(\theta, \varphi) = \begin{cases} N_l^m \cos(m\varphi) P_l^m(\cos \theta) & \text{si } m > 0 \\ N_l^0 P_l^0(\cos \theta) & \text{si } m = 0 \\ N_l^m \sin(-m\varphi) P_l^{-m}(\cos \theta) & \text{si } m < 0 \end{cases} \quad (\text{A.8})$$

A.2 Évaluation efficace d'une combinaison linéaire d'harmoniques sphériques

Nous présentons ici un algorithme simple et efficace d'évaluation d'une combinaison linéaire d'harmoniques sphériques en une direction (θ, φ) donnée. Notons f cette combinaison linéaire :

$$f(\theta, \varphi) = \sum_{l=0}^{L_{\max}} \sum_{m=-l}^l c_{l,m} y_l^m(\theta, \varphi)$$

Plutôt que d'évaluer les valeurs $y_l^m(\theta, \varphi)$ séparément, nous calculons de proche en proche toutes les valeurs des polynômes de Legendre associés selon le schéma suivant : Pour m croissant de 0 à L_{\max} les valeurs $P_m^m(\cos \theta)$ sont calculées à partir de (A.1). Pour chaque m , nous calculons $P_{m+1}^m(\cos \theta)$ en utilisant (A.2) et $P_l^m(\cos \theta)$ grâce à (A.3) pour $m+2 \leq l \leq L_{\max}$.

Nous utilisons également les égalités suivantes pour calculer de proche en proche $\sin(m\varphi)$ et $\cos(m\varphi)$:

$$\begin{aligned} \cos[(n+1)\varphi] &= \cos(n\varphi) \cos \varphi - \sin(n\varphi) \sin \varphi \\ \sin[(n+1)\varphi] &= \sin(n\varphi) \cos \varphi + \cos(n\varphi) \sin \varphi \end{aligned}$$

Chaque couple (l, m) fournit ainsi $y_l^m(\theta, \varphi)$ et $y_l^{-m}(\theta, \varphi)$ en utilisant la définition (A.8). Les constantes de normalisation N_l^m peuvent être calculées de la même façon, ou mieux, pré-calculées et stockées dans un tableau.

Ce processus est détaillé sur les figures A.2 (Sens de calcul) et A.4 (Pseudo-code).

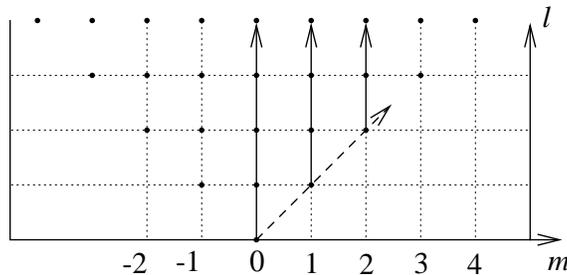


FIG. A.3: Sens de calcul efficace des harmoniques sphériques en (θ, φ) . Pour m allant de 0 à L_{\max} , on calcule y_m^m grâce à (A.1), ainsi que y_{m+1}^m et y_l^m pour $l \geq m+2$ grâce à (A.2) et (A.3)

```

SHComputeArray(integer  $L_{max}$ , float  $\cos_{-}\theta$ , float  $\cos_{-}\varphi$ , float  $\sin_{-}\varphi$ )
   $\cos_{-}m_{-}\varphi = 1$ 
   $\sin_{-}m_{-}\varphi = 0$ 
   $x = \cos_{-}\theta$ 
   $P_{-}m_{-}m = 1$ 

  For  $m$  from 0 to  $L_{max}$ 

     $y_m^m = P_{-}m_{-}m \times N_m^m \times \cos_{-}m_{-}\varphi$ 
     $y_{m+1}^m = x \times (2m+1) \times P_{-}m_{-}m \times \cos_{-}m_{-}\varphi$ 

     $prev\_P = P_{-}m_{-}m$ 
     $curr\_P = x \times (2m+1) \times P_{-}m_{-}m$ 

    From  $l$  from  $m+2$  to  $L_{max}$ 
       $tmp = x \times (2L-1) \times curr\_P - (l-1+m) \times prev\_P$ 
       $prev\_P = curr\_P$ 
       $curr\_P = tmp / (l-m)$ 

       $y_l^m = N_l^m \times curr\_P \times \cos_{-}m$ 

     $P_{-}m_{-}m = -(2m+1) \times \sqrt{1-x^2} \times P_{-}m_{-}m$ 
     $tmp = \cos_{-}m_{-}\varphi \times \cos_{-}\varphi - \sin_{-}m_{-}\varphi \times \sin_{-}\varphi$ 
     $\sin_{-}m_{-}\varphi = \sin_{-}m_{-}\varphi \times \cos_{-}\varphi - \cos_{-}m_{-}\varphi \times \sin_{-}\varphi$ 
     $\cos_{-}m_{-}\varphi = tmp$ 

```

FIG. A.4: Calcul de $y_l^m(\theta, \varphi)$ pour tout $l \leq L_{max}$ et $-l \leq m \leq l$.

A.3 Rotation des harmoniques sphériques

Considérons une harmonique sphérique $y_{l_0}^{m_0}$ exprimée dans un repère orthonormé direct R' . Soit R un autre repère orthonormé direct, de même centre que R' .

Exprimée dans ce nouveau repère, la fonction $y_{l_0}^{m_0}$ n'est plus une harmonique sphérique au sens où nous l'avons défini mais une fonction directionnelle $\tilde{y}_{l_0}^{m_0}$ qui peut cependant, comme toute fonction directionnelle, s'exprimer comme combinaison linéaire d'harmoniques sphériques y_l^m .

La fonction $\tilde{y}_{l_0}^{m_0}$ appartient en fait à une classe plus générale d'harmoniques de degré l_0 [Hob31]. Pour cette raison, il est possible de montrer qu'elle s'écrit plus simplement comme une combinaison linéaire d'harmoniques sphériques de même degré. Nous avons donc :

$$\tilde{y}_{l_0}^{m_0} = \sum_{-l_0 \leq m \leq l_0} \alpha_{l_0, m_0, m} y_{l_0}^m \quad (\text{A.9})$$

avec :

$$\alpha_{l_0, m_0, m} = \int_{\varphi} \int_{\theta} \tilde{y}_{l_0}^{m_0}(\theta, \varphi) y_{l_0}^m(\theta, \varphi) \sin \theta d\theta d\varphi$$

Comme nous le verrons au A.3.2, la connaissance des valeurs $\alpha_{l_0, m_0, m}$ permet de calculer efficacement les coefficients d'harmoniques sphériques d'une fonction directionnelle f exprimée dans R à partir de ceux de f dans R' .

La section suivante traite en particulier du calcul des valeurs $\alpha_{l_0, m_0, m}$.

A.3.1 La formule de rotation

Notons Q l'isométrie qui transforme R' en R . Bien que le calcul direct de $\alpha_{l_0, m_0, m}$ soit extrêmement complexe, le résultat peut être obtenu en décomposant Q en un produit de trois rotations et en calculant les coefficients $\alpha_{l_0, m_0, m}$ pour chaque transformation. Nous composons ensuite ces valeurs pour obtenir $\alpha_{l_0, m_0, m}$ pour Q . Pour cela nous posons :

$$Q^{-1} = R_z(\varphi_0) \circ R_{y'}(\theta_0) \circ R_{z'}(\psi_0)$$

Chaque rotation s'effectue par rapport à un des axes y ou z du repère courant, comme l'indique la figure A.5. Il nous suffit donc de connaître les valeurs $\alpha_{l_0, m_0, m}$ correspondant à ces deux transformations élémentaires.

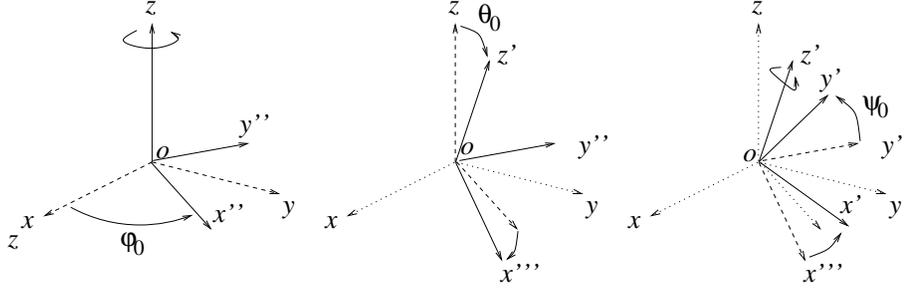


FIG. A.5: Passage de R à R' en trois rotations autour des axes

Rotation autour de l'axe z

Lorsque R se déduit de R' par une simple rotation d'angle φ_0 autour de l'axe z , nous avons :

$$\tilde{y}_{l_0}^{m_0}(\theta, \varphi) = y_{l_0}^{m_0}(\theta, \varphi - \varphi_0)$$

Grâce à (A.8) nous avons immédiatement

$$\tilde{y}_{l_0}^{m_0}(\theta, \varphi) = \cos(m\varphi_0)y_{l_0}^m(\theta, \varphi) - \sin(m\varphi_0)y_{l_0}^{-m}(\theta, \varphi)$$

et donc

$$\alpha_{l_0, m_0, m} = \begin{cases} \cos(m\varphi_0) & \text{si } m = m_0 \\ -\sin(m\varphi_0) & \text{si } m = -m_0 \\ 0 & \text{sinon} \end{cases}$$

Rotation autour de l'axe y

Si R se déduit de R' par une rotation d'angle θ_0 autour de l'axe y , nous avons montré que :

$$\alpha_{l, \varepsilon m_0, \varepsilon m} = S(m) \sin^{\varepsilon_2} \theta_0 \sum_{2L+\varepsilon_1 \leq l} \beta_{l, m_0, |m|, 2L+\varepsilon_1} y_{2L+\varepsilon_1}^0(\theta_0, 0) \quad (\text{A.10})$$

avec :

$$\beta_{l, m_0, m, L} = \frac{2\pi(2l)! N_L^0 N_L^m N_L^{m_0}}{(l-m_0)! l! 2^l} \sum_{(k, p, j, s) \in U_{l, m}} B(k, p, j, s, l, m_0, m) I(k, p, j, s, l, m_0, m)$$

$$U_{l, m} = \left\{ (k, p, j, s) \in \mathbf{N}^4 \left| \begin{array}{ll} 0 \leq 2k \leq m_0 - \varepsilon_3 & 0 \leq p \leq m_0 - 2k - \varepsilon_3 \\ 0 \leq j \leq 2k & 0 \leq s \leq l - m_0 \\ j + p + s + \varepsilon_3 \geq m & j + p + s + \varepsilon_3 + m \text{ even} \end{array} \right. \right\}$$

et :

$$\begin{aligned}
B(k, p, j, s, l, m_0, m) &= \binom{2k}{j} \binom{l-m_0}{s} \binom{m_0-2k-\varepsilon_3}{p} \frac{(-1)^p m_0 2^{m_0-1} (m_0-k-1)! [m(2k-m_0)]^{\varepsilon_3}}{2^{2k} k! (m_0-2k)! [m_0(j+p+s+1)]^{\varepsilon_3}} \\
I(k, p, j, s, l, m_0, m) &= I_1(j+p+s+\varepsilon_3, l-p-j-s-\varepsilon_3, l, m) I_2(p+j+s+\varepsilon_3, m) \\
&\quad \times I_1(m_0-2k-p+j+s-\varepsilon_3-\varepsilon_2, p+l-m_0-s+2k-j, L, 0) \\
I_1(n_1, n_2, l, m) &= \int_0^\pi \sin^{n_1} \theta \cos^{n_2} \theta P_l^m(\cos \theta) d\theta \\
I_2(n, m) &= \int_0^{2\pi} \cos^n \varphi \cos(m\varphi) d\varphi
\end{aligned}$$

Où les quantités $\varepsilon, \varepsilon_1, \varepsilon_2, \varepsilon_3$ et S sont définies par :

$$\left\{ \begin{array}{ll} \varepsilon &= \pm 1 \\ \varepsilon_1 &= 0 \quad \text{si } l+m+m_0 \text{ est pair,} \quad 1 \text{ sinon} \\ \varepsilon_2 &= 1 \quad \text{si } m+m_0 \text{ est pair,} \quad 0 \text{ sinon} \\ \varepsilon_3 &= 0 \quad \text{si } \varepsilon > 0 \quad 1 \text{ sinon} \\ S(m) &= 0 \quad \text{si } m < 0, \quad 1 \text{ sinon} \end{array} \right.$$

Dans le cas particulier correspondant à $m_0 = 0$, l'expression ci-dessus se simplifie et s'écrit :

$$\alpha_{l,0,m} = S(m) \sqrt{2} \sqrt{\frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta_0)$$

Cette expression n'est autre qu'une conséquence directe du théorème d'addition énoncé en A.1.3. Nous avons en effet dans ce cas :

$$\cos \theta' = \sin \theta_0 \sin \theta \cos \varphi + \cos \theta_0 \cos \theta$$

Soit $m \geq 0$. Nous avons alors, en utilisant (A.5) :

$$\begin{aligned}
\alpha_{l,0}^m(\theta_0, 0) &= \int_0^\pi \int_0^{2\pi} y_l^m(\theta, \varphi) \bar{y}_l^0(\theta, \varphi) \\
&= \int_0^\pi \int_0^{2\pi} y_l^m(\theta, \varphi) y_l^0(\theta', \varphi') \\
&= \frac{N_l^0 N_l^m}{\sqrt{2}} \int_0^\pi P_l^m(\cos \theta) \sin \theta \int_{\varphi=0}^{2\pi} \cos(m\varphi) P_l(\cos \theta \cos \theta_0 + \sin \theta \sin \theta_0 \cos \varphi) d\varphi d\theta \\
&= \frac{N_l^0 N_l^m}{\sqrt{2}} 2\pi \frac{(n-m)!}{(n+m)!} P_l^m(\cos \theta_0) \underbrace{\int_0^\pi P_l^m(\cos \theta)^2 \sin \theta d\theta}_{\frac{1}{\pi(N_l^m)^2}} \\
&= \sqrt{2} \sqrt{\frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta_0)
\end{aligned}$$

Le cas $m < 0$ donne immédiatement 0 en remplaçant $\cos(m\varphi)$ par $\sin(m\varphi)$ dans la démonstration de (A.5).

L'expression (A.10) est très intéressante puisqu'elle permet de calculer $\alpha_{l,m_0,m}$ en fonction d'au plus L valeurs d'harmoniques dans le plan $\varphi = 0$. Cela nous permet de calculer rapidement sa valeur pour tout θ_0 pourvu que l'on ait pré-calculé et stocké les constantes $\beta_{l,m_0,|m|,2L+\varepsilon_1}$ intervenant dans leur décomposition.

A.3.2 Décomposition efficace d'une combinaison linéaire d'harmoniques sphériques après rotation

Nous considérons une fonction directionnelle f exprimée dans un repère orthonormé direct R' sous la forme d'une combinaison linéaire finie d'harmoniques sphériques :

$$\forall \theta \quad \forall \varphi \quad f(\theta, \varphi) = \sum_{l=0}^{L_{max}} \sum_{m=-l}^l c_{l,m} y_l^m(\theta, \varphi)$$

Soit R un autre repère orthonormé direct de même centre que R' . Nous nous intéressons au problème de calculer les coefficients de la décomposition en harmoniques sphériques de f exprimée dans R .

Notons \tilde{f} la fonction f exprimée dans R . Nous avons alors, en utilisant les quantités $\alpha_{l,m_0,m}$ introduites en A.3.1.

$$\begin{aligned} \tilde{f}(\theta, \varphi) &= f(\theta', \varphi') \\ &= \sum_{l=0}^{L_{max}} \sum_{m=-l}^l c_{l,m} y_l^m(\theta', \varphi') \\ &= \sum_{l=0}^{L_{max}} \sum_{m=-l}^l c_{l,m} \sum_{m_0=-l}^l \alpha_{l,m_0,m} y_l^{m_0}(\theta, \varphi) \\ &= \sum_{l=0}^{L_{max}} \sum_{m_0=-l}^l \left(\sum_{m=-l}^l c_{l,m} \alpha_{l,m_0,m} \right) y_l^{m_0}(\theta, \varphi) \end{aligned}$$

La décomposition de \tilde{f} en harmoniques sphériques étant unique, nous en déduisons que ses coefficients $\tilde{c}_{l,m}$ valent

$$\tilde{c}_{l,m} = \sum_{m_0=-l}^l c_{l,m_0} \alpha_{l,m_0,m}$$

Comme nous l'avons dit plus haut, les valeurs $\alpha_{l,m_0,m}$ ne sont pas directement calculables pour toute transformation du repère. Nous décomposons donc la transformation de R en R' en trois rotations autour des axes y et z du repère courant et nous composons les coefficients $c_{l,m}$ de proche en proche.

La figure A.6 présente le pseudo-code permettant d'effectuer cette opération.

A.3.3 Coût de l'algorithme

Nous avons implémenté notre algorithme sous la forme d'une librairie permettant de calculer les coefficients $c_{l,m}$ après changement de repère paramétré par les trois angles φ_0, θ_0 et ψ_0 .

Le calcul de $\tilde{c}_{l,m}$ ne nécessite qu'un petit nombre de tableaux temporaires de taille $O(l)$, si bien que le coût mémoire est entièrement dominé par le stockage des valeurs $\beta_{l,m_0,m,L}$. En considérant que $-l \leq m_0, m \leq l$ et $0 \leq l \leq L_{max}$, il y a au plus

$$\sum_{l=0}^{L_{max}} (l+1)(2l+1)^2$$

valeurs à stocker. Cela fait donc un coût mémoire de $O(L_{max}^4)$. Si nous notons N le nombre total d'harmoniques utilisées pour écrire f , ce coût s'écrit $O(N^2)$.

Dans le calcul des nouveaux coefficients $c_{l,m}$ après changement de repère, chaque $\beta_{l,m_0,m,L}$ est utilisé une seule fois et multiplié par une valeur d'harmonique sphérique calculable de proche en proche. Le coût de calcul est donc approximativement $O(L_{max}^4)$ soit $O(N^2)$.

La figure A.7 permet de comparer ce coût avec celui du calcul des nouveaux coefficients par intégration numérique de la fonction tournée, qui est l'alternative naturelle à notre méthode. Nous avons choisi pour cela une fonction dont le spectre des coefficients est bien plein. L'intégration numérique est réalisée par quadrature de Gauss d'ordre 5 sur 10 intervalles en θ et φ quel que soit L_{max} ce qui signifie que la précision du calcul diminue avec L_{max} . Avec ce choix le gain en vitesse est à peu près constant et est d'environ 10^3 . Si

```

Procedure RotateCoefs(angle  $\theta_0, \varphi_0, \psi_0$ )
 $Y[0] = 1$  ,  $Y[1] = \cos \theta_0$ 
 $C_\varphi[0] = C_\psi[0] = 1$ ,  $S_\varphi[0] = S_\psi[0] = 0$ 
 $C_\varphi[1] = \cos \varphi_0$ ,  $C_\psi[1] = \cos \psi_0$ ,  $S_\varphi[1] = \sin \varphi_0$ ,  $S_\psi[1] = \sin \psi_0$ 
For  $l = 1$  to  $l_{max}$ 
  For  $m = -l$  to  $l$ 
    For  $m_0 = 0$  to  $l$ 
       $\alpha = 0$ 
      if  $l + m_0 + m$  is odd then  $\varepsilon_3 = 1$  else 0
      For  $L = 0$  to  $l$  Step 2
        Add  $\beta_{l,m_0,m,L+\varepsilon_3} Y[L]$  to  $\alpha$ 
      If  $m_0 + m$  est impair, multiplier  $\alpha$  par  $\sin \theta_0$ 
      If  $m_0 = 0$  Then
         $\alpha_{l,m_0,m} = \alpha$ 
      Else
        If  $m < 0$  Then
           $\alpha_{l,m_0,m} = \alpha S_\varphi[m_0]$  ,  $\alpha_{l,-m_0,m} = \alpha C_\varphi[m_0]$ 
        else
           $\alpha_{l,m_0,m} = \alpha C_\varphi[m_0]$  ,  $\alpha_{l,-m_0,m} = -\alpha S_\varphi[m_0]$ 
      Done
    Done
     $\tilde{c}_{l,m} = 0$ 
    For  $m_0 = -l$  to  $l$ 
      Add  $(C_\psi[m_0]c_{l,m_0} + S_\psi[m_0]c_{l,-m_0})\alpha_{l,m_0,m}$  to  $\tilde{c}_{l,m}$ 
    Done
  Done
   $Y[l+1] = \frac{1}{l+1} [(2l+1) \cos \theta_0 Y[l] - lY[l-1]]$ 
   $C_\varphi[l+1] = C_\varphi[1]C_\varphi[l] - S_\varphi[1]S_\varphi[l]$ 
   $C_\psi[l+1] = C_\psi[1]C_\psi[l] - S_\psi[1]S_\psi[l]$ 
   $S_\varphi[l+1] = S_\varphi[1]C_\varphi[l] + C_\varphi[1]S_\varphi[l]$ 
   $S_\psi[l+1] = S_\psi[1]C_\psi[l] + C_\psi[1]S_\psi[l]$ 
Done

```

FIG. A.6: Calcul des nouveaux coefficients après une transformation quelconque.

nous adaptons la précision de l'intégration, le gain en vitesse augmente d'autant, puisque notre algorithme de rotation a un coût indépendant de la précision.

Outre sa rapidité, notre méthode a l'avantage de ne pas nécessiter de nouvelles évaluations de la fonction originale. En général ces évaluations sont coûteuses, raison pour laquelle on stocke la dite fonction sous forme d'harmoniques sphériques.

A.3.4 Applications au *clustering* directionnel

L'algorithme de rotation des harmoniques sphériques que nous avons présenté connaît de nombreuses applications, par exemple le design de luminaires ou de réflecteurs. D'une manière générale, il intervient lors de la *combinaison* hiérarchique de fonctions directionnelles exprimées dans des repères différents. C'est par exemple le cas des méthodes de *clustering* dans des environnements non diffus.

Bien que nous n'ayons pas présenté en détail les méthodes hiérarchiques de radiosité dans le cas *non diffus* (voir par exemple [SP94, SAWG91, SDS95]) le lecteur se convaincra facilement des propriétés intuitives suivantes :

- la réflectance des surfaces n'est pas un simple coefficient, mais une fonction bi-

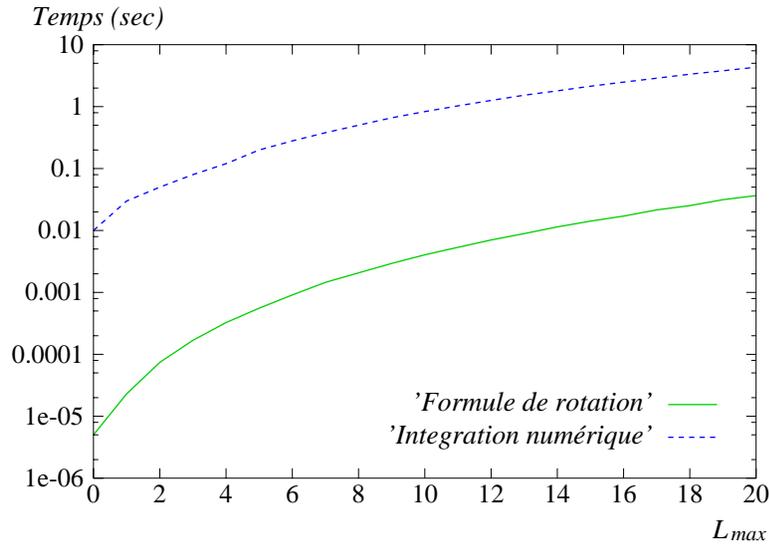


FIG. A.7: Vitesses comparées de calcul des coefficients après rotation en utilisant la formule de rotation et en re-intégrant par quadrature de Gauss. Le gain de vitesse est d'environ 10^3 .

directionnelle traduisant la proportion de la lumière reçue dans une direction et qui repart dans une autre direction (ces fonctions sont appelées BRDF pour *bidirectional reflectance distribution function*).

- chaque surface ne possède pas une unique valeur de radiosit , mais une distribution directionnelle de radiance.

Les harmoniques sph riques sont une base assez bien adapt e   la repr sentation des distributions directionnelles d' nergie lumineuse attach es   chaque surface  l mentaire. Les BRDF  tant connues dans le rep re local de chaque surface, ces distributions de radiance sont  galement exprim es dans le rep re local de chaque surface. En radiosit  hi rarchique, l'op ration de *Push/Pull* consistant   moyenner de mani re ascendante ces distributions peut alors  tre effectu e directement sur les coefficients d'harmonique sph rique des distributions de radiance, puisqu'elles sont toutes exprim es dans le rep re local de la surface, partag e par toutes ses sous-surfaces [SAWG91]. En revanche, dans le cadre du *clustering*, cette propri t  n'est plus v rifi e entre les surfaces d'un m me cluster [SDS95]. Il faut donc dans un premier temps exprimer toutes les distributions de radiance dans un m me rep re, avant de les moyenner. Pour y parvenir, le r - chantillonnage de ces fonctions serait une op ration beaucoup trop co teuse, alors que l'utilisation de la formule de rotation que nous avons pr sent e para t tr s adapt e   ce genre de situations. Dans [SDS95], nous avons  t  limit s par cette difficult , qui est maintenant r solue.

Annexe B

Notes sur la conjecture d'unimodalité

Nous démontrons dans cette annexe, dans un cas particulier, la conjecture d'unimodalité de l'irradiance due à une source convexe. Cette conjecture, qui affirme que l'irradiance due à un émetteur convexe plan admet un unique maximum sur toute droite ou tout plan récepteur, a servi de base à plusieurs méthodes d'encadrement de l'erreur de calcul des facteurs de forme [Hol96, Dre94] et semble à première vue vérifiée, bien que sa démonstration soit pour l'instant inaccessible. Nous en donnons une preuve dans le cas particulier où la source est plane et parallèle au récepteur. Pour cela, nous démontrerons tout d'abord un théorème affirmant que la convolution d'une fonction vérifiant une certaine propriété, par la fonction caractéristique d'un convexe, est unimodale (section B.1). Nous appliquerons ensuite ce théorème à l'irradiance due à une source convexe plane, après nous être ramenés aux conditions du théorème.

B.1 Unimodalité de la convolution

Dans cette section nous démontrons le théorème suivant :

Théorème 4 *Soit une fonction f de classe C^∞ , strictement positive, définie sur \mathbb{R}^2 et vérifiant la propriété suivante :*

Pour tout réel α , $\{ (x, y) \in \mathbb{R}^2 \mid f(x, y) \geq \alpha \}$ est un compact strictement convexe de \mathbb{R}^2 (B.1)

Soit S un compact convexe de \mathbb{R}^2 . Alors la convolution de f par la fonction caractéristique g_S de S vérifie la propriété B.1.

Démonstration

– Notons la convolution G_S et S_y le translaté de S au point y :

$$\forall y \in \mathbb{R}^2 \quad G_S(y) = \int_{S_y} g_S(x) dx$$

– Soient y_1 et y_2 deux points tels que :

$$G_S(y_1) = G_S(y_2) = \gamma$$

Montrons qu'alors :

$$\forall \theta \in [0, 1] \quad G_S((1 - \theta)y_1 + \theta y_2) \geq \gamma$$

Soit $\theta \in [0, 1]$ et $y_\theta = (1 - \theta)y_1 + \theta y_2$. Soit S_1, S_θ et S_2 les translatés de S en y_1, y_θ et y_2 (C.f Figure B.1.a). Soit finalement T la translation de vecteur $y_2 - y_1$ (Nous avons donc $S_2 = T(S_1)$).

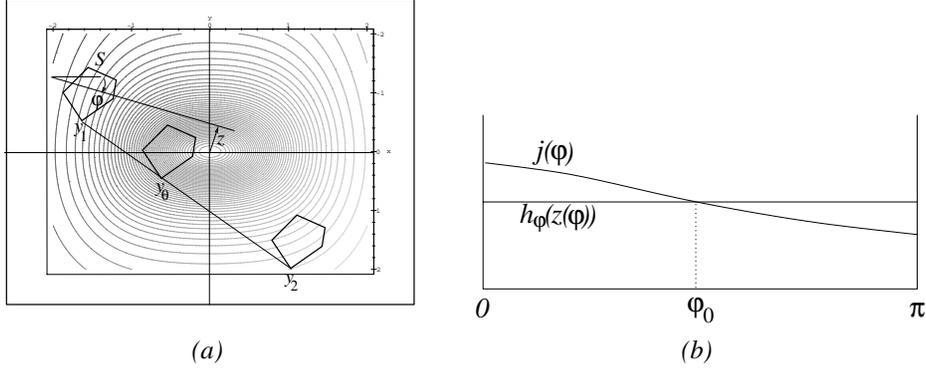


FIG. B.1: Notations pour le théorème d'unimodalité

– Supposons, par l'absurde, que :

$$G_S(y_\theta) < \gamma$$

Nous allons alors construire une suite de convexes compacts emboîtés $(S_1^{(n)})_N$, de mesure non nulle, telle que :

$$\left. \begin{aligned} G_{S_1^{(n)}}(y_1) &= G_{T(S_1^{(n)})}(y_2) = \frac{1}{2^n} \\ G_{S_1^{(n)}}(y_\theta) &< G_{S_1^{(n)}}(y_1) \end{aligned} \right\} \quad (\text{B.2})$$

– Le choix $S_1^{(0)} = S_1$ convient.

– Supposons construit $S_1^{(n)}$ vérifiant (B.2).

– Soit $\varphi \in [0, \pi]$ et $z \in \mathbb{R}$. Soit $E_{\varphi, z}$ le demi-plan défini par :

$$E_{\varphi, z} = \{ (x_1, x_2) \in \mathbb{R}^2 \mid x_1 \cos \varphi + x_2 \sin \varphi < z \}$$

Pour φ fixé, la fonction h_φ définie par :

$$h_\varphi(z) = G_{E_{\varphi, z} \cap S}(y_1)$$

est continue, strictement croissante sur un convexe borné. Son minimum vaut 0, son maximum $G_S(y_1)$. Il existe donc $z \in \mathbb{R}$ tel que :

$$h_\varphi(z) = \frac{1}{2} G_S(y_1)$$

notons le $z(\varphi)$.

– Soit j la fonction définie sur $[0, \pi]$ par (C.f Figure B.1.b) :

$$\forall \varphi \in [0, \pi] \quad j(\varphi) = G_{S \cap E_{\varphi, z(\varphi)}}(y_2)$$

Cette fonction est continue et vérifie par hypothèse :

$$j(0) + j(\pi) = G_{S_1^{(n)}}(y_1)$$

Par continuité, il existe d'autre part $\varphi_0 \in [0, \pi]$ tel que :

$$\begin{aligned} j(\varphi_0) &= \frac{1}{2} (j(0) + j(\pi)) \\ &= G_{S \cap E_{\varphi_0, z(\varphi_0)}}(y_1) \end{aligned}$$

– Nous avons donc :

$$G_{S_1^{(n)} \cap E_{\varphi_0, z(\varphi_0)}}(y_1) = G_{S_1^{(n)} \cap E_{\varphi_0, z(\varphi_0)}}(y_2)$$

Or, par hypothèse, $G_{S_1^{(n)}}(y_\theta) < G_{S_1^{(n)}}(y_1)$. Donc, ou bien :

$$G_{S_1^{(n)} \cap E_{\varphi_0, z(\varphi_0)}}(y_\theta) < G_{S_1^{(n)} \cap E_{\varphi_0, z(\varphi_0)}}(y_1)$$

ou bien :

$$G_{S_1^{(n)} \cap (\mathbb{R}^2 \setminus E_{\varphi_0, z(\varphi_0)})}(y_\theta) < G_{S_1^{(n)} \cap E_{\varphi_0, z(\varphi_0)}}(y_1)$$

Dans le premier cas, nous posons :

$$S_1^{(n+1)} = S_1^{(n)} \cap E_{\varphi_0, z(\varphi_0)}(y_2)$$

dans le second, nous choisissons :

$$S_1^{(n+1)} = S_1^{(n)} \cap (\mathbb{R}^2 \setminus E_{\varphi_0, z(\varphi_0)})$$

– Vérifions que les hypothèses servant la récurrence sont vérifiées par $S_1^{(n+1)}$:
Cet ensemble est convexe, comme intersection de deux convexes. Nous avons bien :

$$G_{S_1^{(n+1)}}(y_1) = \frac{1}{2^{n+1}} \gamma = G_{S_1^{(n+1)}}(y_2) > G_{S_1^{(n+1)}}(y_\theta)$$

et :

$$S_1^{(n+1)} \subset S_1^{(n)}$$

De plus $S_1^{(n+1)}$ est de mesure non nulle car sinon $G_{S_1^{(n+1)}}(y_1) = 0$.

– Posons $R^{(n+1)} = S_1^{(n)} \setminus S_1^{(n+1)}$. Alors :

$$\begin{aligned} \lim_{n \rightarrow \infty} \sum_{k=1}^n G_{R^{(k)}}(y_1) &= G_S(y_1) \sum_{k=1}^{+\infty} \frac{1}{2^k} \\ &= G_S(y_1) \\ &= G_S(y_1) - \lim_{n \rightarrow \infty} G_{S_1^{(n)}}(y_1) \end{aligned}$$

Nous en déduisons que :

$$\lim_{n \rightarrow \infty} G_{S_1^{(n)}}(y_1) = 0$$

Comme f est continue et strictement positive, et que les $S_1^{(n)}$ sont emboîtés nous en déduisons que la limite de la mesure de $S_1^{(n)}$ est 0¹.

Notons maintenant C l'intersection de tous les $S_1^{(n)}$:

$$C = \bigcap_{n=0}^{\infty} S_1^{(n)}$$

La mesure de C est inférieure à celle de tout ensemble $S_1^{(n)}$. Elle est donc nulle. D'autre part, en tant qu'intersection de compacts, C est un compact du plan. L'ensemble C est donc un point ou un segment.

¹En effet, $G_{S_1^{(n)}}(y_1)$ est toujours supérieur à la mesure de $S_1^{(n)}$ fois la valeur minimum de f sur $S_1^{(0)}$

– Cas où \mathcal{C} est un point c .

Notons A_n l'aire de $S_1^{(n)}$. Par continuité de f en c_1, c_θ et c_2 , nous avons :

$$f(c_1) = \lim_{n \rightarrow \infty} \frac{1}{A_n} G_{S_1^{(n)}}(y_1)$$

$$f(c_2) = \lim_{n \rightarrow \infty} \frac{1}{A_n} G_{S_1^{(n)}}(y_2)$$

$$f(c_\theta) = \lim_{n \rightarrow \infty} \frac{1}{A_n} G_{S_1^{(n)}}(y_\theta)$$

Nous avons donc par passage à la limite :

$$f(c_1) = f(c_2) \geq f(c_\theta)$$

C'est absurde car $\theta \in]0, 1[$ et les lignes de niveau de f sont strictement convexes.

– Cas où \mathcal{C} est un segment \overline{cd} .

Par continuité de f , nous avons :

$$\int_{c_1}^{d_1} f(t) dt = \int_{c_2}^{d_2} f(t) dt > \int_{c_\theta}^{d_\theta} f(t) dt \quad (\text{B.3})$$

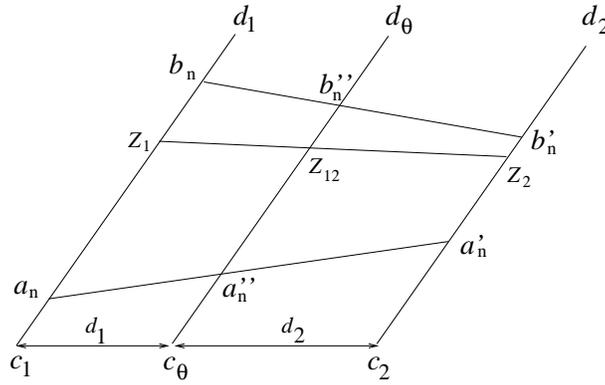


FIG. B.2: Notations

Nous allons construire trois suites d'intervalles emboîtés $([a_n, b_n])_N$, $([a''_n, b''_n])_N$ et $([a'_n, b'_n])_N$ sur chacun des trois segments, telles que (C.f Figure B.2)

$$\int_{a_n}^{b_n} f = \int_{a'_n}^{b'_n} f > \int_{a''_n}^{b''_n} f \quad (\text{B.4})$$

Nous choisissons tout d'abord :

$$a_0 = c_1 \quad b_0 = d_1 \quad a'_0 = c_2 \quad b'_0 = d_2 \quad a''_0 = c_\theta \quad b''_0 = d_\theta$$

D'après (B.3), ces valeurs vérifient (B.4).

Supposons avoir $a_n, b_n, a'_n, b'_n, a''_n, b''_n$ vérifiant (B.4).

Soit :

$$h_1(z) = \int_{a_n}^{a_n + z*(b_n - a_n)} f(t) dt \quad \text{et} \quad h_2(z) = \int_{a'_n}^{a'_n + z*(b'_n - a'_n)} f(t) dt$$

et :

$$h_\theta(z) = \int_{a''_n}^{a''_n + z*(b''_n - a''_n)} f(t) dt$$

Les fonctions h_1 et h_2 sont strictement croissantes sur $[0, 1]$ et valent 0 en 0 et $\int_{a_n}^{b_n}$ en 1.

Il existe donc z_1 et z_2 tels que :

$$h_1(z_1) = h_2(z_2) = \frac{1}{2} \int_{a_n}^{b_n} f(t) dt$$

Soit :

$$z_{12} = \frac{d_2 z_1 + d_1 z_2}{d_1 + d_2}$$

Notons Z_1, Z_2, Z_{12} les points correspondants sur les segments (C.f Figure B.2) D'après les hypothèses de récurrence (B.4), nous nous trouvons dans l'un des états suivants :

– Si $h_\theta(z_{12}) < h_1(z_1)$, alors nous choisissons :

$$a_{n+1} = a_n \quad \text{et} \quad b_{n+1} = a_n + z_1(b_n - a_n)$$

– Si $h_\theta(1) - h_\theta(z_{12}) < h_1(1) - h_1(z_1)$, alors nous choisissons :

$$a_{n+1} = a_n + z_1(b_n - a_n) \quad \text{et} \quad b_{n+1} = b_n$$

Dans les deux cas les valeurs choisies vérifient (B.4). Par récurrence, nous avons également :

$$\forall n \geq 0 \quad \int_{a_n}^{b_n} f = \int_{a'_n}^{b'_n} f = \frac{1}{2^n} \int_{a_0}^{b_0} f$$

En utilisant le même argument que précédemment, les segments $[a_n, b_n]$ et $[a'_n, b'_n]$ convergent vers un segment de longueur nulle. Nous notons c_1 et c_2 les points qui sont leurs limites respectives. En ces points, nous avons

$$f(c_1) = f(c_2)$$

D'autre part, la longueur de $[a''_n, b''_n]$ est toujours bornée, par construction, par celle de $[a'_n, b'_n]$ ou $[a_n, b_n]$. Ce segment converge donc vers un point c_θ . Par passage à la limite, nous avons :

$$f(c_1) = f(c_2) \geq f(c_\theta)$$

C'est absurde car les equipotentiels de f sont strictement convexes.

B.2 Expression de l'irradiance

Nous avons vu en 7.3.6, que l'irradiance due à une source plane définie par son émittance uniforme E et la fonction caractéristique g_S de son support, vaut, dans un plan parallèle à celui de la source, et situé à une distance d de celui-ci :

$$I(y) = \frac{E}{\pi d^2} (g_S * K_d)(y) \quad \text{avec} \quad K_d(x) = \frac{1}{\left(1 + \frac{\|x\|^2}{d^2}\right)^2}$$

Nous observons que le noyau K_d de cette convolution vérifie la propriété (B.1). Cela nous permet d'appliquer le théorème 4. La fonction $y \mapsto I(y)$ vérifie donc à son tour la propriété (B.1). Elle est donc unimodale, car si nous supposons en connaître deux maxima locaux, sa valeur en tout point du segment qui les joint est nécessairement supérieure au minimum de ces deux valeurs, qui n'est alors plus un maximum local.

Bibliographie

- [AC97] Kendall Atkinson and Graeme Chandler. the collocation method for solving the radiosity equation for unoccluded surfaces. *Journal of Integral Equations & Applications*, 1997. To appear (preprint available from <http://www.math.uiowa.edu/atkinson/papers.html>).
- [AH93] L. Aupperle and Pat Hanrahan. A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 155–162, 1993.
- [Arv86] James Arvo. Backward ray tracing, August 1986.
- [Arv94] James Arvo. The irradiance Jacobian for partially occluded polyhedral sources. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 343–350. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [Atk] Kendall E. Atkinson. The planar radiosity equation and its numerical solution. Technical Report 109, Department of Computer Science, University of Iowa, Iowa City, Iowa.
- [Atk89] K. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, New York, Chichester, Brisbane, 1989.
- [ATS94] James Arvo, Kenneth Torrance, and Brian Smits. A Framework for the Analysis of Error in Global Illumination Algorithms. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 75–84, 1994.
- [BA83] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, (4), April 1983.
- [BB84] L. S. Brotman and N. I. Badler. Generating soft shadows with a depth buffer algorithm. *IEEE Computer Graphics and Applications*, 4(10):71–81, October 1984.
- [BDM84] C. Bouville, J. L. Dubois, and I. Marchal. Generating high quality pictures by ray-tracing. In K. Bo and H. A. Tucker, editors, *Eurographics '84*, pages 161–177. North-Holland, 1984.
- [BMSW91] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making Radiosity Usable : Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 51–60, July 1991.
- [BW96] Philippe Bekaert and Yves Willems. Error Control for Radiosity. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 153–164, New York, NY, 1996. Springer-Verlag/Wien.

- [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.
- [CDP95] Frédéric Cazals, George Drettakis, and Claude Puech. Filtering, clustering and hierarchy construction : a new solution for ray tracing very complex environments. In F. Post and M. Göbel, editors, *Computer Graphics Forum (Proc. of Eurographics '95)*, volume 15, Maastricht, the Netherlands, September 1995.
- [CF89] Norman Chin and Steven Feiner. Near real-time shadow generation using BSP trees. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 99–106, July 1989.
- [CG85] Michael F. Cohen and Donald P. Greenberg. The Hemi-Cube : A radiosity solution for complex environments. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 31–40, August 1985.
- [CIE81] CIE. An analytic model for describing the influence of lighting parameters upon visual performance, 1981. Volume 1 : Technical foundations. CIE 19/2.1, Technical Committee 3.1.
- [CPC84] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 137–45, July 1984. Monte Carlo distribution of rays to get gloss, translucency, penumbras, depth of field, motion blur.
- [Cro84] Franklin C. Crow. Summed-area tables for texture mapping. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 207–212, July 1984.
- [DDP97] Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton : A powerful and efficient multi-purpose global visibility tool. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings, Annual Conference Series*, pages 89–100. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [De 97] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings, Annual Conference Series*, pages 361–368. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [DF94] George Drettakis and Eugene Fiume. A Fast Shadow Algorithm for Area Light Sources Using Backprojection. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 223–230, 1994.
- [DKT98] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings, Annual Conference Series*, pages 85–94. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [Dre94] George Drettakis. Structured Sampling and Reconstruction of Illumination for Image Synthesis. CSRI Technical Report 293, Department of Computer Science, University of Toronto, Toronto, Ontario, January 1994.
- [FPSG96] James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A Model of Visual Adaptation for Realistic Image Synthesis. In *Computer Graphics Proceedings, Annual Conference Series, 1996 (ACM SIGGRAPH '96 Proceedings)*, pages 249–258, 1996.
- [GAC⁺88] Andrew S. Glassner, Jim Arvo, Robert L. Cook, Eric Haines, Pat Hanrahan, and Paul Heckbert. Introduction to ray tracing, August 1988.

- [GC93] Steven J. Gortler and Michael F. Cohen. Solving the Radiosity Linear System. In N. M. Thalmann and D. Thalmann, editors, *Communicating with Virtual Worlds : Proceedings of Computer Graphics International '93 (CGI '93)*, pages 78–88, Lausanne, Switzerland, June 1993. Springer-Verlag.
- [Gla84] Andrew S. Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics and Applications*, 4(10) :15–22, October 1984.
- [GS87] Jeffrey Goldsmith and John Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications*, 7(5) :14–20, May 1987.
- [GSCH93] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 221–230, 1993.
- [GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaille. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 212–22, July 1984.
- [Hec90] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 145–154, August 1990.
- [Hec91] Paul Heckbert. *Simulating Global Illumination Using Adaptive Meshing*. Ph.D. thesis, Technical Report, June 1991.
- [Hec92] Paul Heckbert. Discontinuity Meshing for Radiosity. In *Third Eurographics Workshop on Rendering*, pages 203–226, Bristol, UK, May 1992.
- [HH84] Paul S. Heckbert and Pat Hanrahan. Beam tracing polygonal objects. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 119–127, July 1984.
- [HH97] Paul S. Heckbert and Michael Herf. Simulating soft shadows with graphics hardware. Technical report, CS Dept., Carnegie Mellon U., Jan. 1997. CMU-CS-97-104, <http://www.cs.cmu.edu/ph>.
- [Hob31] E. W. Hobson. *The Theory of Spherical and Ellipsoidal Harmonics*. Cambridge University Press, 1931.
- [Hol96] Nicholas Holzschuch. (*Le Controle de l'Erreur dans la Methode de Radiosite Hierarchique (Error Control in Hierarchical Radiosity)*). Ph.D. thesis, Grenoble, France, March 5th, 1996.
- [HS93] Paul Haerberli and Mark Segal. Texture mapping as A fundamental drawing primitive. In Michael F. Cohen, Claude Puech, and Francois Sillion, editors, *Fourth Eurographics Workshop on Rendering*, pages 259–266. Eurographics, June 1993. held in Paris, France, 14–16 June 1993.
- [HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A Rapid Hierarchical Radiosity Algorithm. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.
- [HTSG91] Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg. A Comprehensive Physical Model for Light Reflection. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 175–186, July 1991.
- [HW91a] Eric Haines and John Wallace. Shaft culling for efficient ray-traced radiosity. In *Eurographics Workshop on Rendering*, 1991.
- [HW91b] Paul S. Heckbert and James M. Winget. Finite element methods for global illumination. Technical Report UCB/CSD 91/643, Computer Science Division (EECS), University of California, July 1991.

- [JC98] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 311–320. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [Kaj86] James T. Kajiya. The Rendering Equation. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.
- [Kel97] Alexander Keller. Instant radiosity. In *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, volume 31, 1997. To be published.
- [LSG94] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 67–74. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [LTG93] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 199–208, 1993.
- [Max91] Nelson Max. Unified sun and sky illumination for shadows under trees. *CVGIP : Graphical Models and Image Processing*, 53(3) :223–230, May 1991.
- [MH92] Don Mitchell and Pat Hanrahan. Illumination from Curved Reflectors. In *Computer Graphics (ACM SIGGRAPH '92 Proceedings)*, volume 26, pages 283–291, July 1992.
- [MK94] Karol Myszkowski and Toshiyasu L. Kunii. Texture mapping as an alternative for meshing during walkthrough animation. In *Fifth Eurographics Workshop on Rendering*, pages 375–388, Darmstadt, Germany, June 1994.
- [NDW93] Jackie Neider, Tom Davis, and Mason Woo. *OpenGL Programming Guide*. Addison-Wesley, Reading MA, 1993.
- [Nis98] Tomoyuki Nishita. Light scattering for the realistic rendering of natural scenes. In N. Max and G. Drettakis, editors, *Rendering Techniques '98 (Proceedings of the 9th Eurographics Workshop on Rendering)*, pages 1–10, New York, NY, 1998. Springer-Verlag.
- [NN93] Tomoyuki Nishita and Eihachiro Nakamae. A New Radiosity Approach Using Area Sampling for Parametric Patches. In *Computer Graphics Forum (Eurographics '93)*, volume 12, pages C385–C398, Barcelona, Spain, September 1993.
- [NND96] Tomoyuki Nishita, Eihachiro Nakamae, and Yoshinori Dobashi. Display of clouds and snow taking into account multiple anisotropic scattering and sky light. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 379–386. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.
- [ORDP96] Rachel Orti, Stephane Riviere, Fredo Durand, and Claude Puech. Using the Visibility Complex for Radiosity Computation. In *Lecture Notes in Computer Science (Applied Computational Geometry : Towards Geometric Engineering)*, volume 1148, pages 177–190, Berlin, Germany, May 1996. Springer-Verlag.
- [Ort97] Rachel Orti. *Radiosité Dynamique 2D et Complexe de Visibilité*. Ph.D. thesis, Grenoble, France, juillet 15 1997.
- [Pel95] M. Pellegrini. Monte Carlo Approximation of Form Factors with Error Bounded a Priori. In *Eleventh ACM Symposium on Computational Geometry*, Vancouver, Canada, June 1995.

- [Pho75] Bui-T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6) :311–317, June 1975.
- [PPS97] F. Perez, X. Pueyo, and F. Sillion. Global illumination techniques for the simulation of participating media. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques 97*, pages 309–320. Springer Verlag, Wien, 1997. Proceedings of Eighth Eurographics Workshop on Rendering (Saint-Etienne, France, June 1997).
- [RSC87] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 283–291, July 1987.
- [Sab85] P. Sablonniere. Bernstein-bezier methods for the construction of bivariate spline approximants. *Computer Aided Geometric Design*, 2(1-3) :29–36, 1985.
- [SAG94] Brian Smits, James Arvo, and Donald Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 435–442, 1994.
- [San76] Luis A. Santaló. *Integral Geometry and Geometric Probability*, volume 1 of *Encyclopedia of Mathematics and its applications*. Addison-Wesley, 1976.
- [SAWG91] Francois Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A Global Illumination Solution for General Reflectance Distributions. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 187–196, July 1991.
- [SB86] M. Sweeney and R. Bartels. Ray-tracing free-form B-spline surfaces. *IEEE Computer Graphics and Applications*, 6(2) :41–49, 1986.
- [Sbe93] Mateu Sbert. An Integral Geometry Based Method for Fast Form Factor Computation. In *Computer Graphics Forum (Eurographics '93)*, volume 12, pages C409–C420, Barcelona, Spain, September 1993.
- [SD95] Francois Sillion and George Drettakis. Feature-Based Control of Visibility Error : A Multiresolution Clustering Algorithm for Global Illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 145–152, 1995.
- [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.
- [SH93] Peter Schröder and Pat Hanrahan. On the form factor between two polygons. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 163–164, 1993.
- [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, September 1995. ISSN 1077-2626.
- [SKv⁺92] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, and Paul E. Haeberli. Fast shadows and lighting effects using texture mapping. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 249–252, July 1992.
- [SM92] Brian E. Smits and Gary Meyer. Newton's Colors : Simulating Interference Phenomena in Realistic Image Synthesis. In K. Bouatouch and C. Bouville, editors, *Photorealism in Computer Graphics (Proceedings Eurographics*

- Workshop on Photosimulation, Realism and Physics in Computer Graphics, 1990*), pages 185–194, 1992.
- [Sol95] Cyril Soler. Représentation multi-échelles d'informations de visibilité, June 1995. Mémoire de DEA de Mathématiques Appliquées. Université Joseph Fourier. Grenoble.
- [Sol96] Cyril Soler. Dot products between spherical harmonics. Technical report, iMAGIS/GRAVIR-IMAG, september 1996.
- [SP94] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann publishers, San Francisco, 1994.
- [SS95] Peter Schroder and Wim Sweldens. Spherical Wavelets : Efficiently Representing Functions on the Sphere. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 161–172, 1995.
- [SS96a] Cyril Soler and François Sillion. Caractérisation multi-échelles de la visibilité pour les calculs de radiosit . *Revue Internationale de CFAO et d'Informatique Graphique*, 11(4), 1996.
- [SS96b] Cyril Soler and François Sillion. Accurate Error Bounds for Multi-Resolution Visibility. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 133–142, New York, NY, 1996. Springer-Verlag/Wien.
- [SS98a] Cyril Soler and François Sillion. Automatic calculation of soft shadow textures for fast, high-quality radiosity. In *Ninth Eurographics Workshop on Rendering*, Vienna, Austria, 1998.
- [SS98b] Cyril Soler and François X. Sillion. Fast calculation of soft shadow textures using convolution. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings, Annual Conference Series*, pages 321–332. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [SSS97] M. Stamminger, Ph. Slusallek, and H.-P. Seidel. Bounded radiosity – illumination on general surfaces and clusters. In D. Fellner and L. Szirmay-Kalos, editors, *Proc. EUROGRAPHICS '97*. Blackwell, 1997.
- [Sta98] Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings, Annual Conference Series*, pages 395–404. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [Tam92] Filippo Tampieri. Accurate form-factor computation. In David Kirk, editor, *Graphics Gems III*, pages 329–333. Academic Press, San Diego, 1992. includes code.
- [TH93] Seth Teller and Pat Hanrahan. Global visibility algorithms for illumination computations. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 239–246, 1993.
- [TM93] Roy Troutman and Nelson L. Max. Radiosity algorithms using higher order finite element methods. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 209–212, 1993.
- [VFGP93] A. Vesel, M. Feda, M. Gervautz, and W. Purgathofer. Radiosity computation in flatland using a monte carlo approach with extended rays. In *Proceedings of International Conference on Computer Graphics (ICCG '93)*, pages 181–194, June 1993. held in Budmerice, Slovakia.
- [Wan92] Leonard Wanger. The effect of shadow quality on the perception of spatial relationships in computer generated imagery. In David Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, pages 39–42, March 1992.

- [War94] Gregory J. Ward. A contrast-based scalefactor for luminance display. In Paul S. Heckbert, editor, *Graphics Gems IV*. Academic Press, San Diego, 1994.
- [Wat90] Mark Watt. Light-Water Interaction Using Backward Beam Tracing. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, pages 377–385, August 1990.
- [WEH89] John R. Wallace, Kells A. Elmquist, and Eric A. Haines. A ray tracing algorithm for progressive radiosity. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 315–324, July 1989.
- [Whi80] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6) :343–349, June 1980.
- [Wil78] Lance Williams. Casting curved shadows on curved surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, volume 12, pages 270–274, August 1978.
- [WPF90] Andrew Woo, Pierre Poulin, and Alain Fournier. A survey of shadow algorithms. *IEEE Computer Graphics and Applications*, 10(6) :13–32, November 1990.
- [Zat93] Harold R. Zatz. Galerkin radiosity : A higher-order solution method for global illumination. In *Computer Graphics Proceedings, Annual Conference Series : SIGGRAPH '93 (Anaheim, CA, USA)*, pages 213–220. ACM SIGGRAPH, New York, August 1993.