

2m11.2835.5

Université de Montréal

Construction et utilisation de bloqueurs
pour l'accélération des requêtes de visibilité

par

Luc Leblanc

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

août 2000

© Luc Leblanc, 2000



2 2580 14-05

Université de Montréal

Construction et installation de plusieurs
pour l'acquisition des techniques de visibilité

QA
76
1154
2000
N.045

par
le

Département de génie civil et de génie des matériaux
École Polytechnique

Document communiqué à la suite des démarches effectuées

par le service de l'information de la bibliothèque

pour le service de la bibliothèque

de l'Université de Montréal

2000

1000-1000-1000



Université de Montréal
Faculté des études supérieures

Ce mémoire de maîtrise intitulé

Construction et utilisation de bloqueurs
pour l'accélération des requêtes de visibilité

présenté par
Luc Leblanc

a été évalué par un jury composé des personnes suivantes :

Président : Victor Ostromoukhov

Directeur de recherche : Pierre Poulin

Membre : Max Mignotte

Sommaire

Un grand nombre de techniques en image de synthèse ont recours à des requêtes de visibilité et peuvent grandement bénéficier de leur accélération. Que ce soit pour effectuer l’affichage en temps réel lors du *parcours* d’une scène ou bien pour capter le moindre effet de lumière lors du calcul de l’*illumination globale*, les requêtes de visibilité représentent la majorité du calcul de ces processus.

Parmi les différentes techniques permettant d’accélérer la vitesse de ces requêtes, une catégorie d’algorithmes éliminent une bonne portion de la scène non visible en utilisant des *bloqueurs*. Ainsi, toute la géométrie cachée par ces bloqueurs est rapidement exclue des calculs.

Ce mémoire aborde le problème de la visibilité en image de synthèse, et plus particulièrement celui de l’élimination rapide des surfaces par l’utilisation de bloqueurs. Dans un premier temps, un processus de création est proposé pour trois différents types de bloqueurs : planaires, directionnels et volumiques. Les bloqueurs planaires permettent de représenter l’*occlusion* provoquée par de grandes régions planaires tels les murs, planchers, tables et autres. La définition et la création de ce type de bloqueurs est par la suite étendue en un tout nouveau concept plus général, soit celui de bloqueur directionnel. Ce nouveau type de bloqueurs permet de tenir compte de l’occlusion provoquée par des surfaces courbes telles les dossiers de chaises et les rideaux. Finalement, pour les objets fermés, un processus de création de bloqueurs volumiques est introduit.

Par la suite, un tout nouvel algorithme de visibilité est introduit en *radiosité hiérarchique par agrégats* faisant appel aux bloqueurs pour tirer profit de la visibilité complète et de l’occlusion complète. Ainsi, les échanges d’énergie effectués entre deux surfaces ou régions d’une scène complètement bloquées par d’autres surfaces sont détectées à un stage très tôt de l’algorithme. Les échanges complets, c’est-à-dire sans l’occlusion d’aucune autre surface, sont aussi détectés rapidement. L’efficacité réelle de la technique est ensuite démontrée sur une série variée de scènes allant d’un niveau de détails simple à complexe.

Ce nouvel algorithme de visibilité est par la suite présenté dans le cadre du calcul de l’*illumination directe* provoquée par des sources lumineuses étendues lors du *rendu* d’une image par *tracé de rayons*. La méthode proposée est ainsi démontrée comme étant

générale, simple et efficace.

Mots clefs : bloqueur planaire, bloqueur directionnel, bloqueur volumique, occlusion total, visibilité complète, radiosité, tracé de rayons.

Table des matières

Remerciements	vii
1 Introduction	1
1.1 Motivation	2
1.2 Contribution	2
1.3 Survol du travail	3
2 Travaux antérieurs	4
2.1 Visibilité en général	4
2.1.1 Visibilité approximative	5
2.1.2 Visibilité exacte	6
2.1.3 Visibilité conservatrice	6
2.2 Bloqueurs en visibilité	7
3 Les bloqueurs	10
3.1 Bloqueurs planaires	12
3.1.1 Construction d'une hiérarchie d'agrégats de faces	13
3.1.2 Extraction de grands bloqueurs	19
3.2 Bloqueurs directionnels	25
3.2.1 Définition	27
3.2.2 Création	27
3.3 Bloqueurs volumiques	32
3.3.1 Création de bloqueurs internes	33
4 Radiosité	36
4.1 Description de la radiosité	36

4.1.1	Survol de la radiosité	37
4.1.2	Techniques implantées	47
4.2	Description de la technique de visibilité	52
4.2.1	Visibilité complète	53
4.2.2	Occlusion totale	55
4.2.3	Système de visibilité	57
4.3	Résultats et analyse	59
4.3.1	Statistiques sur nos scènes test	59
4.3.2	Analyse	60
4.4	Extension	63
4.4.1	Bloqueurs directionnels	63
4.4.2	Bloqueurs volumiques	66
5	Lumières étendues	69
5.1	Description du problème	69
5.2	Description de la méthode	70
5.2.1	Technique standard	70
5.2.2	Technique avec bloqueur	70
5.3	Résultats	70
5.4	Améliorations	71
5.4.1	Regroupement en agrégats de lumières	71
5.4.2	Subdivision de lumières	73
5.4.3	Regroupement par surfaces ou par régions d'image	73
6	Conclusion	74
7	Glossaire	76
	Bibliographie	79

Liste des tableaux

4.1	Visibilité complète et occlusion totale sur des scènes variées de la figure 4.13	62
4.2	Visibilité complète et occlusion totale sur des scènes variées de la figure 4.13	66
5.1	Visibilité complète et occlusion totale sur les différentes scènes de la figure 5.1	71

Table des figures

3.1	Deux points de vue d'un divan et de ses agrégats de faces à un certain niveau de la hiérarchie de faces	13
3.2	Hiérarchie d'agrégats de faces	14
3.3	Graphe dual	15
3.4	Cône de normales	16
3.5	Différentes étapes de l'extraction de bloqueurs	23
3.6	Lors de l'élargissement du bloqueur, ce dernier s'accroche sur le contour (en haut à droite) et ne peut plus gonfler. Diminuer un peu le bloqueur permet d'améliorer la situation	24
3.7	Grands bloqueurs convexes extraits en semi-transparent apposés sur le maillage original de la scène	25
3.8	La scène originale ainsi que les bloqueurs	25
3.9	Les rideaux coupent complètement la transmission de lumière entre la scène et la salle	26
3.10	Différentes vues d'un bloqueur directionnel	28
3.11	Boîte englobante d'un agrégat	29
3.12	Différentes étapes de la construction d'un bloqueur directionnel	31
3.13	Enveloppe interne	34
3.14	Bloqueur volumique opaque (1482 triangles) et modèle original transparent (40000 triangles)	35
4.1	a) scène originale; b) scène après discrétisation en éléments	39
4.2	a) fonction originale; b) base constante; c) base linéaire	40
4.3	Étapes du processus de résolution par radiosit�	41
4.4	Calcul du rapport hauteur/largeur d'un triangle	42

4.5	a) Sommet en T ; b) vue en 3D ; c) une solution au problème	43
4.6	Facteur de forme et ses composantes	43
4.7	Facteur de forme entre une surface différentielle et un polygone de n côtés	44
4.8	a) base constante ; b) reconstruction par interpolation Gouraud	47
4.9	Étapes du processus de subdivision d'un triangle (les flèches pleines représentent les bonnes étapes et la flèche pointillée représente une mau- vaise subdivision provoquant des triangles allongés)	52
4.10	Étapes de la création d'un tunnel. a) Agrégats de faces et leurs boîtes englobantes ; b) Ajout de la boîte englobante du tunnel ; c) Ajout de plans définissant le tunnel	54
4.11	Le tunnel formé entre deux groupes d'éléments est testé pour occlusion totale. L'agrégat du haut bloque partiellement le tunnel, tandis que celui du bas le bloque totalement puisqu'il intersecte tous les segments.	56
4.12	La densité du maillage est plus élevée aux alentours de l'émetteur	63
4.13	Scènes test	64
4.14	Résultat de différents facteurs d'amplitude α sur une scène	67
4.15	Scènes test	68
5.1	Scènes test	72

Liste des Algorithmes

1	Construction de la hiérarchie d'agrégats de faces	16
2	CréerHiérarchie(ListeBoîte): Construction de la hiérarchie d'agrégats de volumes	20
3	Extraction de bloqueurs rectangulaires	26
4	Ajustement de la pyramide de visibilité	32
5	Radiosité par propagation	46
6	Système de résolution de radiosité hiérarchique par propagation	49
7	TransporteLumière(p, q)	49
8	PushPull(p, \hat{B}_{bas})	49
9	Oracle(p, q)	50
10	Oracle2(p, q, ref(v))	58

Remerciements

J'aimerais remercier ma famille pour leur support lors de ces deux années de travail. Je voudrais remercier mon directeur de recherche, Pierre Poulin, avec qui j'ai apprécié travailler, tant pour ses compétences dans le domaine de l'infographie qui m'ont aidé à produire ce mémoire que pour son support constant. À tous les membres du laboratoire que j'ai côtoyé tout au long de la maîtrise, merci pour les nombreuses discussions techniques et aussi pour leur amitié qui m'ont permis de passer de bons moments. Merci à Filippo Tampieri pour les discussions techniques que nous avons eues.

Finalement je voudrais remercier le CRSNG et Discreet pour leur contribution financière.

Chapitre 1

Introduction

Dans différents domaines mais principalement dans celui de la synthèse d'images les requêtes de visibilité revêtent une grande importance. En effet, lors du rendu d'images, il est nécessaire de déterminer les parties visibles de la scène du point de vue de la caméra. De plus, pour simuler différents effets de lumière, la visibilité des sources à la scène doit être calculée, et même entre les surfaces de la scène pour traiter l'éclairage indirect. De même, pour parcourir une scène virtuelle, la visibilité occupe une importance de premier plan.

De nombreux algorithmes ont vu le jour et ont permis de solutionner correctement le problème des requêtes de visibilité. Malgré cela, de nouveaux algorithmes, ceux de ce mémoire inclus, continuent d'être introduits sur la base de ce problème. La raison, fort simple, est que la détermination de la visibilité demeure dans de nombreux algorithmes le coût majeur pouvant ainsi représenter plus de 99% du temps de calcul. C'est pourquoi tant de méthodes sont introduites encore aujourd'hui, tentant de diminuer l'ordre de grandeur de l'évaluation des requêtes de visibilité.

Deux différents types de requêtes existent : les requêtes de type "blocage" et celles de type "détermination de surfaces visibles". Dans le premier cas, on ne cherche pas à savoir ce que l'on voit mais plutôt si deux points (ou deux régions) sont mutuellement visibles. Dans le second cas, on veut savoir exactement quelles surfaces sont visibles d'un point ou d'une région. Pour accélérer ces requêtes, des techniques conservatrices permettent d'éliminer une bonne partie de la scène réduisant ainsi l'espace de travail. Plusieurs de ces techniques font appel à des bloqueurs pour éliminer rapidement les surfaces cachées.

Ce mémoire traite de la construction de tels bloqueurs et de leurs utilisations pour accélérer les requêtes de type “blocage”, bien qu’ils puissent être utilisés avec succès dans le cas de la détermination de surfaces visibles. L’efficacité de la méthode est plus particulièrement démontrée en radiosité hiérarchique et lors du calcul de l’éclairage direct de sources étendues en tracé de rayons.

1.1 Motivation

Bien que plusieurs techniques ont recours aux bloqueurs, très peu d’entre elles se préoccupent de les construire. Elles prennent pour acquis que les surfaces (très souvent des polygones) présentes dans la scène sont directement utilisables pour fournir beaucoup d’occlusion. C’est le cas pour les scènes simples possédant peu de détails et de grandes surfaces. Toutefois la plupart des scènes ont un niveau de détails très élevé et sont construites à l’aide d’un très grand nombre de petites surfaces. Malgré ce fait que chaque surface individuellement ne provoque que peu d’occlusion, leur agrégation quant à elle peut démontrer un bon taux de blocage. C’est pour pouvoir tenir compte de ce type d’occlusion que l’on présente des algorithmes permettant de construire à partir de triangles des bloqueurs planaires, directionnels et volumiques.

Lors du calcul de l’éclairage direct ou indirect, que ce soit en radiosité, en tracé de rayons ou par lignes de balayage, il existe plusieurs types de cohérence entre les différentes requêtes de visibilité. Ainsi, deux points rapprochés d’une même région de l’espace possèdent généralement une fonction de flux lumineux très similaire. De plus, la visibilité d’un point au reste de la scène est très similaire pour des directions rapprochées. Ces différents cas peuvent bénéficier d’accélération mettant à profit leur cohérence propre. Les bloqueurs introduits au cours de ce mémoire utiliseront cette cohérence afin de diminuer le coût des requêtes.

1.2 Contribution

Plusieurs contributions sont apportées par ce mémoire. Tout d’abord, nous présentons une nouvelle façon de créer des bloqueurs planaires à partir d’une “soupe de triangles” grâce à l’agrégation de faces. Ensuite, ce type de bloqueurs est généralisé sous une nouvelle forme, les bloqueurs directionnels, permettant de tenir compte de

l'occlusion provoquée par des surfaces quelconques mais continues. Enfin, l'efficacité de ces deux types de bloqueurs est démontrée par l'introduction d'un nouvel algorithme de visibilité en radiosité. L'efficacité et l'adaptabilité sont aussi démontrées dans le contexte du calcul de l'éclairage provenant de sources étendues.

Toutefois, il est important de noter que malgré les bonnes performances apportées par ces techniques, tous les problèmes ne sont pas résolus. Ainsi, l'occlusion provoquée par de multiples bloqueurs disjoints n'est pas traitée. Par exemple, une feuille d'un arbre ne bloque presque rien mais le feuillage complet en bloque davantage, et une forêt, bien plus encore. De plus, certaines scènes ou éléments de scène ne provoquent pas ou peu d'occlusion et ne peuvent ainsi profiter de ce type d'accélération. On retrouve parmi ces exemples, les média participatifs tels les nuage, le brouillard et d'autres types d'objets comme les grillages et les clôtures.

1.3 Survol du travail

Ce mémoire débute par un bref survol des techniques de visibilité. Plus d'espace est accordé aux techniques récentes d'occlusion et de création de bloqueurs. Lors du chapitre suivant (chapitre 3) la création de trois types de bloqueurs est présentée. Les chapitres 4 et 5 portent sur l'utilisation de ces bloqueurs dans différents contextes. Dans le chapitre 4, une nouvelle technique de visibilité utilisant les bloqueurs est expliquée pour la radiosité, puis pour le calcul de l'éclairage direct au chapitre 5. Finalement, les aspects positifs et les limitations des techniques présentées sont discutés lors de la conclusion.

Chapitre 2

Travaux antérieurs

La détermination de la visibilité est un des problèmes fondamentaux en informatique graphique. Les différents aspects de ce problème sont donc adressés dans un très large éventail de publications. Par souci de concision, seul un très bref survol de ces techniques est fait au cours de ce chapitre. Pour de plus amples détails, les lecteurs intéressés peuvent se référer à des recueils plus complets [Dur99, Gra92, WPF90, FvFH90, SSS74]. Dans la première section, la visibilité en général est abordée. Puis, les techniques récentes d'occlusion sont décrites avec plus de détails dans la seconde section.

2.1 Visibilité en général

On peut classer de bien des façons les nombreux algorithmes de visibilité. Certains déterminent les surfaces visibles ou potentiellement visibles d'un point de vue précis ou d'une région ; d'autres déterminent si deux points ou régions de l'espace sont mutuellement visibles ou partiellement visibles. Plusieurs techniques sont basées sur un point de vue ponctuel (*pinhole camera*), et par conséquent elles sont plus difficilement applicables pour déterminer de façon spécifique la visibilité entre paires d'éléments comme c'est le cas dans un algorithme de radiosité.

Nous avons choisi de présenter les différentes techniques de visibilité en les divisant selon les catégories suivantes : visibilité approximative, visibilité exacte et visibilité conservatrice.

2.1.1 Visibilité approximative

Une importante classe de méthodes de visibilité est basée sur l'échantillonnage, que cela soit pour déterminer les surfaces visibles [App68] ou bien calculer un pourcentage d'occlusion [CPC84]. Parmi ces dernières, la plus populaire [WEH89, HSA91] consiste simplement à lancer un certain nombre de rayons entre une paire d'éléments, et à faire correspondre la visibilité à la fraction des rayons ayant atteint leur objectif sans blocage. Ces méthodes sont simples, peuvent être efficaces et leur précision observée est normalement augmentée en utilisant plus d'échantillons, au coût additionnel du temps de calcul. Diverses techniques existent pour accélérer le *lancer de rayons*. Ainsi, les grilles régulières [FK85, AW87], les *arbres octaires* [Gla84] et les hiérarchies de volumes englobants [RW80, Smi98] permettent de sélectionner rapidement les objets à intersecter. Cependant, ces méthodes souffrent du manque de bornes sur l'erreur de la visibilité estimée, conduisant ainsi dans certaines configurations géométriques à des fuites de lumière ou d'ombre. Une fuite de lumière peut apparaître lorsque par exemple de la lumière se glisse le long d'un mur non éclairé. Dans la situation inverse, il s'agit d'une fuite d'ombre.

La visibilité volumique [Sil94, Sil95] associe une densité à une représentation volumique d'un ensemble de surfaces, et l'occlusion est approximée comme un facteur d'atténuation. Bien que cette technique fournit un sens général de l'occlusion produite, elle suppose une distribution uniforme de petits objets et par conséquent, échoue à reproduire fidèlement les occlusions de nature plus directionnelle ou nette. Néanmoins, à partir de ce modèle il est possible de traiter des médias participatifs tels la fumée dans le calcul de l'éclairage indirect par radiosité.

Plusieurs techniques reposent sur la construction d'images pour accélérer différentes requêtes de visibilité. La *carte d'ombre* de Williams [Wil78] est un *tampon de profondeur* construit à partir d'une lumière ponctuelle pour calculer l'ombre provoquée par cette dernière sur la scène. La carte d'ombre peut être étendue aux sources non ponctuelles à l'aide des *LDI* (multi-couches de profondeurs) [SGHS98, ARHM00]. Chrysanthou *et al.* [CCOL98] proposent de construire une structure en cinq dimensions se servant d'images pour accélérer le calcul de l'éclairage. La précision de ces algorithmes dépend de la résolution des images (tampons) construites, mais cependant la taille de la mémoire requise pour ces structures augmente aussi considérablement.

2.1.2 Visibilité exacte

Beaucoup de techniques classiques pour déterminer les surfaces visibles d'un point de vue retournent un résultat exact mais à une précision fixée à l'avance par la taille de l'image désirée¹. Ces techniques comprennent entre autres le tampon de profondeur [Cat74, GK93], le balayage de lignes [Bou70], l'algorithme du peintre par *arbre BSP* [FKN80] et les algorithmes par subdivision d'aire [War69, WA77].

Les techniques de visibilité analytique 3D [PD90, DF94, SG94, DDP97] fournissent la solution complète et précise au problème de la visibilité. Tous les types d'événements de visibilité dans une scène sont étudiés. Il s'agit des combinaisons *sommet-arête* et *arête-arête-arête* de tous les polygones de la scène. Malheureusement, la complexité algorithmique de ces techniques devient rapidement un souci sérieux pour des scènes très complexes². Même si certaines extensions [DDP99] en font une technique plus pratique en diminuant la complexité du problème, leur robustesse demeure toujours un grave problème sans réelle solution.

2.1.3 Visibilité conservatrice

Les techniques mentionnées jusqu'à maintenant sont souvent inefficaces lorsque la scène contient un nombre élevé d'objets. Tout un groupe d'algorithmes dits conservateurs tentent d'accélérer les différentes requêtes de visibilité en diminuant rapidement l'ensemble de travail initial. Ces algorithmes retournent un ensemble de surfaces possiblement visibles d'un point de vue ou d'une région. Pour être efficaces, elles doivent retourner le moins de surfaces possible qui ne sont pas réellement visibles. De plus, pour être considérées conservatrices, il est essentiel que "toutes" les surfaces visibles soient contenues dans l'ensemble final.

Certains de ces algorithmes sont utilisés à l'exécution d'un autre algorithme de visibilité, mais plusieurs servent à produire un *PVS* (*potentially visible set*) [Air90, TS91, COFHZ98, WBP98], c'est-à-dire une subdivision de l'espace en cellules pour chacune desquelles un ensemble de surfaces possiblement visibles est assigné.

Les *portals* [TH93] offrent un gain majeur entre les éléments distribués dans plu-

¹La précision peut être supérieure à la taille de l'image si le travail s'effectue sur un tampon plus grand.

²L'ordre de grandeur de ces algorithmes va jusqu'à atteindre $O(n^9)$ où n est le nombre de surfaces contenues dans la scène.

sieurs pièces séparées par des ouvertures. En fait, les éléments contenus dans deux pièces séparées se voient si et seulement si, il existe un ensemble visible d'ouvertures (les portails) entre eux. La technique détermine toutes les pièces que voit une pièce par ces ouvertures et ce, pour toutes les pièces de la scène. Si l'information à propos des portails n'est pas incluse dans la structure de la scène, des méthodes [TH93, Men98] existent pour construire une telle structure de façon automatique. Cependant, les portails deviennent inefficaces pour tout ce qui diffère des scènes architecturales typiques, pour la visibilité à l'intérieur d'une pièce et aussi lorsque la visibilité mutuelle entre pièces augmente.

Deux techniques de base et très employées de concert permettent de réduire efficacement la taille de la scène. La pyramide de vue élimine toutes les surfaces hors d'une pyramide formée généralement de quatre plans latéraux (définis par l'image) et parfois de deux autres plans définissant les limites avant et arrière. Dans la seconde technique, toutes les faces arrière³ peuvent être éliminées, diminuant presque de moitié la scène.

Un dernier type d'élimination de surfaces utilise l'occlusion provoquée par les surfaces rapprochées afin d'éliminer celles derrière. Certaines de ces méthodes [ZMHH97, GK93] opèrent en espace image et peuvent ainsi utiliser sans restriction n'importe quel type de surface comme bloqueur. D'autres [Pla93, CT97, HMC⁺97] opèrent en espace objet et se limitent à l'utilisation de grands bloqueurs convexes. Elles peuvent éliminer une importante partie de toute la scène lors de parcours dans une scène. Cependant, on doit fournir une liste de bloqueurs qui dépend de la position la caméra ou des paires d'éléments en radiosité. De plus, lorsque la scène est surtout constituée d'une soupe de petits polygones, ces techniques ne peuvent fusionner ces petits bloqueurs pour en former de plus grands qui eux seraient plus utiles. Dans de telles situations, le gain attendu est réduit de façon importante.

2.2 Bloqueurs en visibilité

De récentes techniques pour construire des bloqueurs afin de calculer le PVS d'une scène ont été introduites, certaines en 2D [KCCO00, WWS00], et d'autres en 3D [DDTP00, SDDS00]. Ces algorithmes utilisent la fusion de bloqueurs afin d'augmenter

³Pour tous les objets fermés, les faces faisant dos au point de vue sont considérées cachées par les faces avant.

l'efficacité de l'occlusion réussissant ainsi à éliminer beaucoup de surfaces cachées.

Koltun *et al.* [KCCO00] construisent des bloqueurs virtuels définis comme des objets convexes⁴ représentant l'occlusion à partir d'une cellule de façon conservatrice. Pour construire un de leurs bloqueurs, ils choisissent un objet puis fabriquent un bloqueur situé à l'arrière, tout en étant conservateur à partir de la cellule. Ensuite, ils cherchent un autre objet leur permettant d'allonger le bloqueur. Pour ce faire, ils utilisent un objet intersectant une des deux lignes de support formées par la cellule et le bloqueur. Le nouveau bloqueur composé sera situé à l'arrière de tous les objets utilisés pour former l'occlusion. Ainsi, objet par objet, le bloqueur allonge jusqu'à ce qu'il n'y ait plus d'objets pour améliorer l'occlusion. Bien que cette technique permette d'éliminer beaucoup de surfaces, elle demeure difficilement extensible en 3D, limitant ainsi son utilité.

Pour construire leurs bloqueurs, Wonka *et al.* [WWS00] se servent de l'observation suivante : l'occlusion provoquée à partir d'un point de vue par un bloqueur rétréci par un epsilon demeure valide (conservateur) pour tout point de vue contenu à l'intérieur d'un rayon de epsilon du premier point de vue. Ainsi, pour calculer la visibilité d'une cellule, ils échantillonnent son contour et utilisent la technique de rétrécissement de bloqueurs pour diminuer les objets pouvant provoquer beaucoup d'occlusion. Cette dernière technique peut s'étendre en 3D mais demande beaucoup de points d'échantillonnage, ce qui augmente considérablement le coût du temps de calcul.

Durand *et al.* [DDTP00] calculent la visibilité d'une cellule en projetant dans un premier temps des bloqueurs potentiels sur un plan de façon sous-estimée⁵ formant ainsi une image, puis en comparant l'occlusion avec des objets qui sont surestimés⁶ dans le but d'être conservateurs. Donc si un objet surestimé est complètement caché par l'image des bloqueurs sous-estimés, cet objet est invisible à partir de la cellule. Pour obtenir une image d'occlusion finale (servant de bloqueur), les bloqueurs sont projetés sur différents plans à l'aide d'un opérateur de projection étendue, puis sont reprojétés sur un nouveau plan par convolution (basé sur l'idée de Soler et Sillion [SS98]) pour pouvoir les combiner ensemble.

Schauffler *et al.* [SDDS00] convertissent leur scène en représentation volumique

⁴Puisqu'ils se limitent au cas 2D, leurs bloqueurs sont en réalité des segments de ligne.

⁵Par sous-estimée on entend que la taille des bloqueurs est réduite pour tenir compte de façon conservatrice de la visibilité à partir d'une cellule.

⁶Par surestimés on entend que la taille des objets est augmentée.

(arbre octaïdre séparant intérieur, extérieur et frontière). Pour obtenir la visibilité d'une cellule, ils construisent des tunnels à l'arrière de leurs bloqueurs volumiques représentant les zones cachées (analogue aux ombres d'une source étendue). Les bloqueurs choisis sont dans un premier temps des cellules intérieures qui sont allongées tant que cela est possible : le bloqueur reste à l'intérieur d'un objet ou encore, entre dans un tunnel d'ombre (fusion de bloqueurs). L'allongement des bloqueurs arrête lorsque ces derniers atteignent la frontière d'un objet ou d'un tunnel d'ombre. Puisque le contenu d'un tunnel d'ombre est invisible de la cellule, les bloqueurs peuvent s'étendre dans cette zone tout en demeurant conservateur. Seuls les objets qui ne sont pas contenus dans les tunnels d'ombre sont considérés visibles par la cellule. Le problème de cette méthode est qu'elle exige que les bloqueurs soient convertis dans une telle représentation volumique, ce qui n'est pas toujours possible. Certaines configurations peuvent aussi exiger un très grand niveau de subdivision de l'arbre octaïdre.

D'autres méthodes récentes ont été présentées pour extraire de grands bloqueurs à partir d'un modèle polygonal. Une de ces approches utilise la simplification de *maillages* tout en préservant l'occlusion [LT99]. Leurs bloqueurs ne sont pas convexes, caractéristique essentielle à la technique présentée dans ce mémoire pour garantir l'occlusion complète. Une autre approche gonfle des cellules d'un arbre octaïdre à l'intérieur d'un modèle géométrique fermé [ASVN00]. Ce dernier algorithme est restreint aux modèles fermés, ce qui n'est souvent pas garanti dans une grande quantité de *modèles* commerciaux. Une technique d'extraction présentée dans ce mémoire est similaire à celle-ci, mais basée sur la surface et non pas le volume.

Chapitre 3

Les bloqueurs

Lors du rendu, du calcul de l'illumination directe ou de l'illumination globale d'une scène, une des principales étapes pour le réalisme, et typiquement la plus coûteuse de toutes, est sans contredit le calcul de la visibilité. La complexité de ce calcul augmente selon les besoins mentionnés précédemment. Ainsi, le calcul de l'illumination globale est le plus coûteux de ces calculs puisqu'il exige de connaître les événements de visibilité entre chaque paire de surfaces de la scène et non seulement entre les surfaces et la caméra comme c'est le cas lors du rendu, ou entre les surfaces et les lumières lors du calcul de l'illumination directe.

Comme la complexité du calcul varie en fonction du nombre de surfaces et qu'il est généralement très élevé (jusqu'à $O(n^2)$ pour n surfaces), alors il est très important de diminuer le nombre de surfaces sur lequel on effectue le travail. Pour ce faire, des techniques de visibilité conservatrice éliminent rapidement une grande proportion des surfaces à l'aide de tests généralement triviaux et peu coûteux. Certaines des techniques mentionnées dans le chapitre 2 utilisent l'*élimination de surfaces par pyramide de vue*, consistant à éliminer toutes les surfaces exclues d'une pyramide formée de six plans. De plus récentes techniques réussissent à éliminer encore plus de surfaces en tenant compte de l'occlusion provoquée par de grandes surfaces. Ces techniques, dites techniques par *élimination de surfaces par occlusion*, utilisent donc des bloqueurs pour éliminer des surfaces cachées dites, les *bloqués*.

La grande majorité de ces techniques utilisent les polygones inclus dans la scène comme bloqueurs, faisant l'hypothèse que certains d'entre eux seront de bons bloqueurs. Toutefois, de plus en plus, les scènes actuelles comportent un niveau de détails

très élevé résultant en de nombreux petits polygones. Ainsi, ces scènes contiennent souvent très peu de “bons” bloqueurs polygonaux. On se retrouve donc devant la nécessité d’extraire ou plutôt de construire un ensemble de bloqueurs à partir des surfaces (dans notre cas des polygones) de la scène initiale. Malheureusement, très peu de travaux ont porté sur ce sujet, et ce mémoire se propose d’introduire et d’analyser une telle approche.

Dans le présent chapitre, la construction de trois différents types de bloqueurs sera présentée : les bloqueurs planaires, directionnels et volumiques. Ces trois types de bloqueurs devront posséder un certain nombre de caractéristiques pour qu’ils soient considérés de bons bloqueurs.

La première et principale caractéristique que tous les bloqueurs doivent posséder est qu’un bloqueur soit conservateur, c’est-à-dire qu’il ne doit en aucun temps bloquer de la géométrie par rapport à un point de vue ou une région de l’espace que la scène originale ne bloque pas. Ceci permet de ne pas introduire d’erreurs dans le processus de visibilité.

Les bloqueurs doivent être grands pour éliminer le plus de bloqués possible. C’est pourquoi on cherche à trouver lors de la construction des bloqueurs, les plus grandes régions intérieures et les plus grandes surfaces.

La troisième caractéristique est la simplicité du bloqueur. Le but des techniques d’élimination de surfaces par occlusion est d’éliminer le plus de surfaces possible et surtout le plus rapidement possible pour diminuer le coût des autres techniques de visibilité qui devront ensuite être appliquées. Pour ce faire, les bloqueurs doivent être simples pour que le test d’occlusion effectué à l’aide du bloqueur soit rapide.

Finalement, on exige que les bloqueurs soient convexes puisque la plupart des techniques les utilisant ainsi que les nouvelles méthodes présentées dans les chapitres suivants nécessitent cette dernière caractéristique pour accomplir correctement et efficacement leurs tâches¹.

¹La dernière technique présentée pour construire des bloqueurs volumiques ne rencontre pas cette dernière caractéristique.

3.1 Bloqueurs planaires

Lorsqu'on importe des scènes de différents systèmes de modélisation, il n'est pas rare de n'avoir que très peu d'information géométrique de celles-ci. On ne connaît pas leurs subdivisions spatiales, si les objets sont fermés², les adjacences des surfaces, ou même la hiérarchie de ses objets. On se retrouve donc souvent avec ce qu'on appelle communément une "soupe de polygones", c'est-à-dire un mélange quelconque de polygones avec aucune autre information que celle de chaque polygone individuel : réflectance, émission, normale, coordonnées spatiales et de texture, etc.

Sans perte de généralité, on peut substituer une scène de polygones planaires quelconques par une scène contenant exclusivement des triangles. En effet, on peut toujours trianguler un polygone planaire quelconque. Par conséquent dans le reste de ce document, on ne considérera que des triangles. Partant de cet ensemble comme entrée initiale à notre problème, on tentera d'extraire de grands bloqueurs planaires reposant sur les surfaces (ensemble de triangles) de la scène. Les bloqueurs ainsi produits seront donc un sous-ensemble des surfaces de la scène et par le fait même, ce type de bloqueurs possédera la première caractéristique essentielle d'être conservateur.

Le choix d'extraire dans un premier temps ce type de bloqueurs repose sur trois faits. Très peu d'informations sont nécessaires pour les fabriquer et on en dispose justement de très peu. Ce type de bloqueurs simple à construire puis à tester répond ainsi à une autre des caractéristiques voulues. Il est à noter que la forme du bloqueur extrait doit être simple (par exemple très peu de côtés) pour répondre correctement au critère. Finalement, dans certains cas comme pour des régions planaires sans volume, ce type de bloqueurs est le plus approprié, pour ne pas dire la meilleure solution.

Pour extraire les bloqueurs de la scène initiale, il faut localiser les grandes régions planaires contenues dans celle-ci. Pour ce faire, un algorithme d'agrégation de faces, dans notre cas de triangles, sera utilisé afin d'obtenir de grandes régions à partir des triangles initiaux en établissant l'information de connexité. Cet algorithme, introduit par Garland [Gar99], sera décrit en détail dans la prochaine section. Du moment où les régions seront créées, il faudra les localiser puis par la suite, l'extraction des bloqueurs se fera sur chacune des régions choisies tel qu'expliqué dans une section subséquente

²Un objet fermé est un objet pour lequel il existe une zone intérieure bien définie séparée complètement de la zone extérieure.

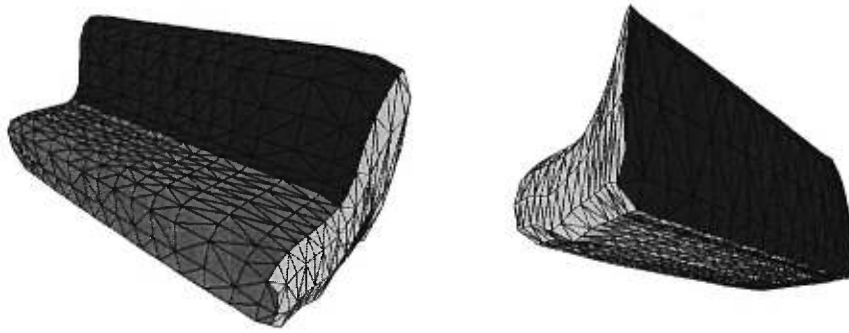


FIG. 3.1: Deux points de vue d'un divan et de ses agrégats de faces à un certain niveau de la hiérarchie de faces

(3.1.2).

3.1.1 Construction d'une hiérarchie d'agrégats de faces

L'agrégation de faces [WHG99] a été introduite en radiosit  hi rarchique par agr gats dans le but de produire un meilleur transfert d' nergie entre agr gats de polygones formant des r gions surfaciques presque planaires, tout en gardant le m me ordre de complexit  que les algorithmes conventionnels d'*agr gats de volumes* [Sil94].

Un *agr gat de faces* se d finit comme un ensemble de triangles inter-connect s formant une seule surface, mais pouvant toutefois contenir des trous. L'agr gat de faces contient aussi un ensemble de caract ristiques associ es   la r gion, entre autres l'aire, la normale moyenne, le *c ne de normales* [SA93], les bo tes englobantes orient e et align e sur les axes, etc. La figure 3.1 montre deux points de vue d'un divan et de ses agr gats de faces (  un certain niveau de la hi rarchie) identifi s par diff rentes teintes.

  partir d'une sc ne initiale, on construit une hi rarchie d'agr gats de faces. Dans une telle hi rarchie, les feuilles de l'arbre correspondent aux triangles et les noeuds   des agr gats de faces form s   leur tour d'un certain nombre d'agr gats de faces. Dans ce qui suit, le nombre d'enfants d'un noeud est toujours deux, par cons quent l'arbre construit est binaire. Le haut de l'arbre (la racine et ses quelques descendants) est g n ralement form  par des agr gats de volumes puisque   un certain point, il n'est plus possible de joindre deux agr gats de faces pour en construire une nouvelle r gion inter-connect e. La figure 3.2 montre un exemple d'une hi rarchie binaire et compl te d'agr gats de faces.

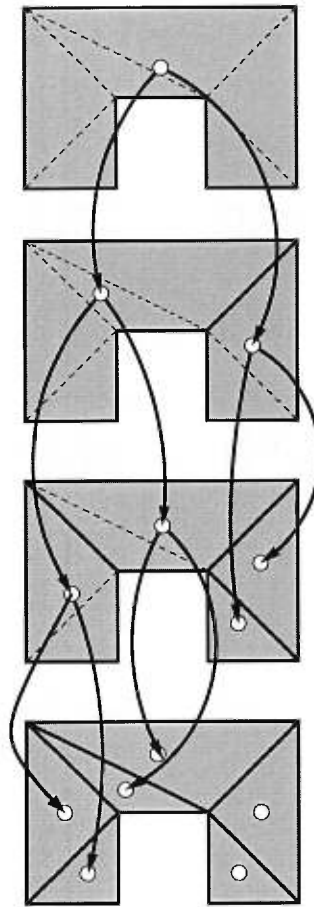


FIG. 3.2: Hiérarchie d'agrégats de faces

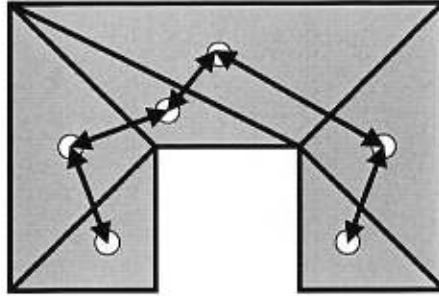


FIG. 3.3: Graphe dual

Algorithme

La construction de la hiérarchie d'agrégats de faces est un processus itératif vorace pendant lequel on joint deux à deux des agrégats pour en former un nouveau. Initialement chaque triangle de la scène produit un agrégat ne contenant que les caractéristiques de celui-ci. Pour décider quels agrégats joindre pour en former de nouveaux, un graphe dual est créé où chaque noeud correspond à un agrégat. Les arêtes duales du graphe ne joignent que les agrégats voisins, c'est-à-dire partageant au moins une arête de leurs triangles. Ainsi les régions formées seront toutes connectées. Un exemple de tel graphe est présenté à la figure 3.3. À chaque arête duale est associé un coût permettant ainsi d'ordonner les fusions d'agrégats pour obtenir des régions répondant aux critères désirés. Dans la prochaine section les métriques permettant de calculer le coût sont présentées en détail.

Pour construire un nouveau noeud de la hiérarchie, il suffit de choisir l'arête duale restante dans le graphe avec le plus petit coût. Après avoir formé le nouvel agrégat, toutes les arêtes duales associées à l'un ou l'autre des agrégats enfants sont reliées au nouveau noeud, puis leur coût est révisé pour tenir compte de la fusion. Ce processus de construction vorace est décrit en pseudo-code par l'algorithme 1.

Lors de la construction d'un nouvel agrégat de faces, plusieurs données rattachées à celui-ci sont calculées comme son aire et son périmètre. Entre autres, un cône de normales [SA93] est calculé puis associé à l'agrégat. Ce cône comme illustré à la figure 3.4, est défini par sa normale moyenne et un angle. Il représente l'ensemble des normales associées à la surface et permet d'estimer rapidement le degré de planarité de la surface. Différentes techniques existent pour le calculer mais toutes demandent de traiter les n normales composant la surface. Toutefois, le cône peut dans notre cas

Algorithme 1: Construction de la hiérarchie d'agrégats de faces

```

Données : triangles de la scène
Résultat : hiérarchie d'agrégats de faces

pour chaque triangle faire construire un agrégat de faces;
pour chaque paire de triangles voisins faire
    créer une arête duale;
    calculer le coût de l'arête;
    insérer l'arête dans le monceau;
fin
tant que Taille(monceau) > 0 faire
    retirer l'arête duale ayant le plus petit coût;
    créer un agrégat contenant les deux agrégats de faces liés par l'arête;
    mettre à jour les arêtes duales (coûts et connexions);
fin

```

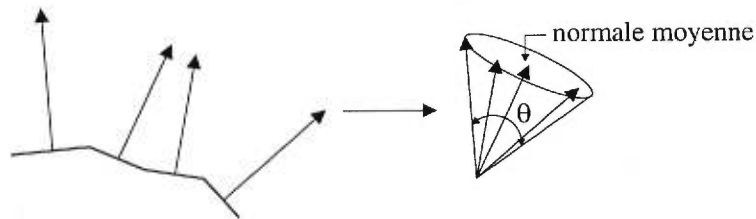


FIG. 3.4: Cône de normales

être approximé de façon conservatrice à partir des deux cônes de normales associés à chacun des agrégats enfants, accélérant ainsi le coût du calcul.

Métrique

La forme des agrégats de faces ainsi que l'entière hiérarchie dépendent de la métrique utilisée pour assigner un coût à chacune des arêtes duales. Cette métrique doit être formulée de telle façon à s'assurer que les régions formées satisfassent un certain nombre de critères. La première caractéristique recherchée est la planarité. Une région ne doit pas être nécessairement plane, mais plus son cône de normales est petit (la variation de ses normales), meilleure est la région. Cette caractéristique permet d'extraire de plus grands bloqueurs, mais aussi une meilleure solution de radiosité comme il sera expliqué lors du prochain chapitre. La forme des régions est aussi très importante. Elle doit être la plus compacte et régulière que possible, c'est-à-dire tendre vers la forme du cercle. Ceci a pour effet de produire des bloqueurs couvrant un grand angle solide. De même la solution de radiosité utilisant ces agrégats sera plus précise.

La métrique présentée par Garland [Gar99] utilise dans un premier temps des quadriques pour évaluer la planarité d'une région, puis un second facteur pour tenir compte

de la forme.

Ainsi pour calculer la planarité d'un agrégat, on utilise la distance au carré au plan moyen de tous ses sommets. Ceci est exprimé dans l'équation 3.1 où \mathbf{v}_i représente chacun des k sommets et (\mathbf{n}, d) est le plan moyen. Après quelques transformations algébriques (équations 3.1 à 3.5), on peut réécrire cette distance sous la forme d'une sommation de quadriques se combinant en une seule (équation 3.5). Une quadrique Q (équations 3.6 et 3.7) se compose d'une matrice \mathbf{A} , d'un vecteur \mathbf{b} et d'un scalaire c .

$$E_q = \frac{1}{k} \sum_i (\mathbf{n}^T \mathbf{v}_i + d)^2 \quad (3.1)$$

$$= \frac{1}{k} \sum_i (\mathbf{n}^T \mathbf{v}_i + d)(\mathbf{v}_i^T \mathbf{n} + d) \quad (3.2)$$

$$= \frac{1}{k} \sum_i (\mathbf{n}^T \mathbf{v}_i \mathbf{v}_i^T \mathbf{n} + 2d \mathbf{v}_i^T \mathbf{n} + d^2) \quad (3.3)$$

$$= \frac{1}{k} \sum_i (\mathbf{n}^T (\mathbf{v}_i \mathbf{v}_i^T) \mathbf{n} + 2\mathbf{v}_i^T d \mathbf{n} + d^2) \quad (3.4)$$

$$= \frac{1}{k} \sum_i Q_i(\mathbf{n}, d) = \frac{1}{k} \left(\sum_i Q_i \right) (\mathbf{n}, d) \quad (3.5)$$

où

$$Q_i = (\mathbf{A}_i, \mathbf{b}_i, c_i) = (\mathbf{v}_i \mathbf{v}_i^T, \mathbf{v}_i, 1), \quad (3.6)$$

$$Q(\mathbf{n}, d) = \mathbf{n}^T \mathbf{A} \mathbf{n} + 2\mathbf{b}^T (d \mathbf{n}) + cd^2. \quad (3.7)$$

Pour obtenir la quadrique d'un agrégat de faces, on pourrait extraire tous ses sommets puis l'évaluer, mais ceci deviendrait une solution coûteuse pour l'ensemble des agrégats. Une autre solution, proposée par Garland [Gar99], est de calculer la quadrique de chaque triangle initial. Lors de la formation d'un nouvel agrégat, sa quadrique se définit comme la somme des quadriques de ses enfants. Cette technique s'intègre bien dans le cadre de l'algorithme de création d'agrégats de faces, n'est pas très coûteuse en mémoire (une quadrique par agrégat) et se calcule rapidement. Le seul problème est que la quadrique calculée de cette façon n'est pas tout à fait la même. En effet, lors de la combinaison des quadriques, les sommets partagés par chacun des agrégats se trouvent à être dédoublés. Malgré cette différence, l'utilisation de cette quadrique

demeure une très bonne métrique pour évaluer efficacement la planarité d'une région.

La quadrique telle que définie par l'équation 3.5 permet d'évaluer la distance moyenne au carré du plan moyen d'un agrégat. Mais pour obtenir cette distance, il faut tout d'abord qu'un tel plan soit défini. Ce plan minimisant la quadrique peut être calculé par les techniques classiques d'analyse de la composante principale [Jol86] où la matrice \mathbf{Z} de covariance des sommets est construite. Le vecteur propre ayant la plus petite valeur propre de la matrice \mathbf{Z} représente la normale du plan. Cette matrice de covariance peut se définir en fonction de la quadrique :

$$\mathbf{Z} = \mathbf{A} - \frac{\mathbf{b}\mathbf{b}^T}{c}. \quad (3.8)$$

Le second terme de la métrique sert à évaluer la forme de l'agrégat. L'irrégularité γ d'un agrégat est définie grâce à son périmètre ρ et son aire A tel que :

$$\gamma = \frac{\rho^2}{4\pi A}. \quad (3.9)$$

Un cercle possède un irrégularité de 1 et plus cette valeur est grande, moins l'agrégat est compact. L'erreur de forme (équation 3.10) indique le coût de joindre deux agrégats en un nouveau.

$$E_f = 1 - \frac{\max(\gamma_1, \gamma_2)}{2\gamma} \quad (3.10)$$

où γ_1 est l'irrégularité du premier agrégat,
 γ_2 est l'irrégularité du second agrégat,
 γ est l'irrégularité du nouvel agrégat.

Cette métrique permet d'obtenir de très bons résultats sur les surfaces courbes mais fonctionne moins bien sur les surfaces planaires. En fait, sur ce type de surface la métrique a pour effet de favoriser la fusion de grandes surfaces compactes avec de petites surfaces puisque l'irrégularité demeure bonne. Pour favoriser la fusion d'agrégats de tailles semblables pour former une hiérarchie mieux balancée, la métrique peut être améliorée en la multipliant par l'aire du nouvel agrégat :

$$E_f = 1 - \frac{\max(\gamma_1, \gamma_2)}{2\gamma} (A_1 + A_2)\sigma \quad (3.11)$$

- où A_1 est l'aire de premier agrégat,
 A_2 est l'aire du second agrégat,
 σ est une constante de normalisation assurant que $(A_1 + A_2)\sigma < 1$.

Agrégats de volumes

La construction de la hiérarchie d'agrégats de faces se termine lorsqu'il ne reste que des régions disjointes ou bien, lorsque le coût pour en former de nouvelles est considéré trop élevé. On obtient alors une liste de hiérarchies d'agrégats de faces. Pour obtenir une hiérarchie unique, on regroupe les différentes hiérarchies d'agrégats de faces en agrégats de volumes jusqu'à l'obtention d'une racine unique contenant la scène en entier. Pour ce faire, de nombreuses techniques existent dont certaines [GH96, Sil95] utilisant les boîtes englobantes de l'arbre octaire de toute la scène. D'autres encore, utilisent plutôt un *arbre k-d* pour former leurs agrégats de volumes, générant ainsi un arbre mieux balancé.

Mueller *et al.* [MSF99] présentent une technique simple et efficace qui produit une hiérarchie compacte et bien distribuée. La technique (voir pseudo-code de l'algorithme 2) consiste à ordonner la liste des boîtes englobantes selon leurs centres le long des trois axes de la scène. On évalue ensuite le coût associé à diviser la liste en deux sections après chaque élément de la liste, et ce pour chaque dimension. La fonction de coût de diviser une liste H (généralement H est une liste d'arbres) en deux sous-listes H_{gauche} et H_{droite} est la suivante :

$$C_H(axe) = \frac{S(H_{gauche})}{S(H)} |H_{gauche}| + \frac{S(H_{droite})}{S(H)} |H_{droite}| \quad (3.12)$$

- où $|H|$ est le nombre d'objets dans la hiérarchie H ,
 $S(H)$ est la surface de la boîte englobante associée à H ,
 $axe \in \{X, Y, Z\}$.

3.1.2 Extraction de grands bloqueurs

La section précédente a présenté en détail les algorithmes nécessaires pour fabriquer une hiérarchie binaire d'une scène à partir d'une soupe de triangles. Avec cette hiérarchie d'agrégats de faces et de volumes, on possède maintenant suffisamment d'information sur la connexité, la planarité et la forme des régions pour être capable d'extraire de cette géométrie un ensemble de grands bloqueurs convexes. Dans un premier

Algorithme 2: CréerHiérarchie(ListeBoîte) : Construction de la hiérarchie d'agrégats de volumes

Entrée : ListeBoîte : liste de boîtes englobantes
Sortie : une boîte englobante
Résultat : hiérarchie binaire de boîtes englobantes

Liste[0] \leftarrow ListeBoîte triée en x ;
 Liste[1] \leftarrow ListeBoîte triée en y ;
 Liste[2] \leftarrow ListeBoîte triée en z ;
 CoûtMin $\leftarrow \infty$;

pour $j \leftarrow 0$ à 2 **faire**
 pour $i \leftarrow 1$ à Taille(Liste[j]) - 1 **faire**
 Coût \leftarrow coût de séparation de Liste[j] à la position i ;
 si Coût < CoûtMin **alors**
 CoûtMin \leftarrow Coût ;
 Position $\leftarrow i$;
 MeilleureListe $\leftarrow j$;
 fin
fin
fin

ListeGauche \leftarrow sous-liste formée des éléments 1 à Position de Liste[MeilleureListe];
 ListeDroite \leftarrow sous-liste formée des éléments Position + 1 à Taille(Liste[MeilleureListe]) de Liste[MeilleureListe];
 BoîteGauche \leftarrow CréerHiérarchie(ListeGauche);
 BoîteDroite \leftarrow CréerHiérarchie(ListeDroite);
 retourner Boîte englobant BoîteGauche et BoîteDroite

temps, il faut identifier les grandes régions planaires pouvant produire beaucoup d'occlusion. Pour chaque région choisie, on extrait un petit nombre de bloqueurs convexes, lesquels seront insérés dans une hiérarchie pour accès rapide.

Identification des régions planaires

On considère une région comme pouvant produire de bons bloqueurs lorsque celle-ci est grande et planaire³. Parmi ces régions on retrouve des éléments de scènes tels les planchers, les plafonds, les murs, les portes, les tables, les étagères, et encore bien d'autres.

Pour identifier ces régions, on utilise un algorithme classique de descente en profondeur de la hiérarchie d'agrégats. De cette façon, on s'assure que les régions identifiées sont effectivement les plus grandes possible puisque la région contenue dans un noeud est l'union de celles de ses enfants. On débute le parcours à la racine de la hiérarchie d'agrégats de la scène, vérifiant si les agrégats de faces rencontrés sont suffi-

³Une région non planaire peut aussi fournir de très bons bloqueurs mais ces régions seront traitées par les bloqueurs directionnels et volumiques décrits plus loin au cours de ce chapitre.

samment grands et planaires. Les agrégats de volumes sont tout simplement traversés récursivement jusqu'à l'obtention d'agrégats de faces. La planarité d'un agrégat de faces est vérifiée à l'aide du cône de normales calculé lors de la construction de l'agrégat.

Si un agrégat de faces ne satisfait pas aux critères, on continue à descendre la hiérarchie jusqu'à ce que l'algorithme trouve un bon agrégat ou que le noeud atteint soit trop petit pour être considéré comme pouvant fournir de bons bloqueurs. La taille minimale d'un bloqueur est spécifiée par l'utilisateur sous forme d'une fraction de la taille de la scène. Ainsi, si une région tombe sous ce seuil, l'algorithme peut arrêter sa descente dans cette section de la hiérarchie.

Toutes les régions situées en périphérie de la scène, tels les planchers et les plafonds dans certaines scènes, ne peuvent bloquer aucun élément à cause de leurs positions. Ces régions ne sont pas utilisées pour en extraire des bloqueurs. Il est à noter que d'autres techniques plus complexes pourraient être utilisées afin de déterminer toutes les régions ne bloquant rien ou très peu de géométrie. Toutefois, la pertinence de telles techniques n'est pas très élevée puisque le coût engendré serait sans doute élevé. Il est à noter que les bloqueurs sont insérés dans une hiérarchie binaire et que par conséquent, l'accès se fait en temps logarithmique. Ainsi cette petite diminution du nombre de bloqueurs engendrée par de telles techniques n'affecterait que très peu le rendement final des bloqueurs.

Extraction des bloqueurs à partir des agrégats de faces

Pour chaque agrégat de faces satisfaisant l'ensemble des critères ci-haut, on doit extraire un ensemble restreint de polygones convexes planaires (les bloqueurs) approximant de façon conservatrice, mais le plus complet possible, l'occlusion de l'agrégat. Les bloqueurs extraits résident donc sur le plan de support de la région.

Comme il le sera démontré plus loin, puisque la technique d'extraction repose en partie sur la construction d'un *arbre quaternaire*, la taille et l'emplacement des bloqueurs dépendent de l'orientation de la boîte englobante de la région. Le système d'axes de l'arbre est aligné sur cette dernière. Il est alors capital de trouver une bonne boîte englobante de l'agrégat traité. Utiliser la boîte englobante ayant la plus petite aire possible [GS87, Gla89] semble résulter en l'extraction d'excellents bloqueurs dans toutes les scènes testées.

Pour minimiser l'aire de la boîte, différentes techniques peuvent être utilisées. Une approche simple et naïve consiste à discrétiser les orientations par tranches fixes. Elle permet de trouver une bonne solution à la précision voulue. Toutefois, pour améliorer le résultat dans les scènes architecturales courantes (généralement alignées sur les axes), on débute la discrétisation sur un des axes de la scène. Ceci a pour effet de permettre de trouver la solution exacte pour toutes les régions étant effectivement alignées sur les axes. D'autres techniques plus complexes permettent de trouver la solution exacte au problème. Par exemple, Evans *et al.* [EST⁺95] présentent un algorithme en temps linéaire pour trouver le parallélogramme minimal englobant une région convexe. Combiné à un algorithme calculant l'*enveloppe convexe* d'un ensemble de points [Ede87], on obtient une boîte d'aire minimale.

Le périmètre de l'agrégat de faces est extrait et traduit dans le système de coordonnées du plan de support formant ainsi un polygone 2D. Le polygone quelconque obtenu peut être concave et même contenir des trous. L'extraction de bloqueurs se fait en 2D sur ce polygone. Pour obtenir le périmètre de la région on peut se servir de la hiérarchie d'agrégats de faces formant cette dernière. En effet, il suffit simplement de partir des feuilles de l'arbre, puis d'extraire le contour de chaque triangle. Par la suite, on fusionne les contours au fur et à mesure que l'on remonte la hiérarchie jusqu'à la région. Lors de la fusion de deux contours, le périmètre est obtenu en unissant les arêtes des deux sous-sections, puis en retirant de l'ensemble toutes les arêtes dédoublées⁴.

Le reste de notre algorithme d'extraction suit une démarche similaire à l'algorithme d'extraction de bloqueurs volumiques de Andujar *et al.* [ASVN00], sauf que l'on traite des polygones 2D au lieu de volumes. Comme on n'a pas d'information sur nos modèles a priori, certains modèles peuvent être troués ou planaires, comme les murs et les planchers. On ne peut utiliser l'approche des bloqueurs volumiques dans ces cas. C'est pourquoi, l'extraction de bloqueurs surfaciques est essentielle.

On peut en premier lieu vérifier si le polygone est convexe et déjà simple, c'est-à-dire possédant peu d'arêtes. Si c'est le cas, l'extraction est inutile puisqu'on est en présence d'un bon bloqueur. Alors on n'a seulement qu'à le retourner, sinon on procède à l'extraction.

⁴Les deux arêtes dédoublées sont retirées et non pas une seule.

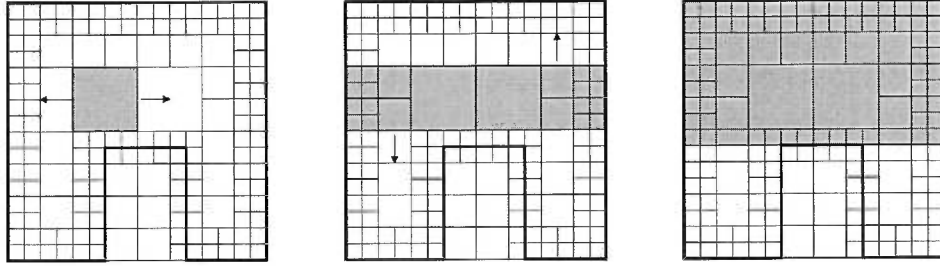


FIG. 3.5: Différentes étapes de l'extraction de bloqueurs

Comme le démontre l'algorithme 3, après avoir obtenu le périmètre de la région, on l'utilise pour construire un arbre quaternaire contenant notre polygone 2D. Chaque cellule de l'arbre est marquée comme étant soit intérieure, extérieure ou sur le contour du polygone. Dans ce dernier cas, la liste des arêtes est ajoutée à la cellule. Pour déterminer si une cellule est interne ou externe, on intersecte en 2D avec le polygone un rayon partant du centre de la cellule se dirigeant à l'infini dans une direction quelconque. Si la cardinalité de l'ensemble des arêtes intersectées est paire, la cellule est considérée externe et dans le cas inverse, la cellule est interne.

À partir de cet arbre quaternaire, l'extraction de bloqueurs de forme rectangulaire se fait de manière itérative à l'aide d'un algorithme vorace à germe. Une cellule interne au polygone, le germe, est choisie puis gonflée jusqu'à s'appuyer sur le contour de la région (voir figure 3.5). De nouveaux germes seront choisis pour créer de nouveaux bloqueurs jusqu'à ce que la superficie du polygone soit bien couverte. De façon plus détaillée, les étapes sont les suivantes :

1. Choisir la cellule intérieure non marquée⁵ la plus grande de l'arbre quaternaire comme étant le bloqueur initial.
2. Élargir ce bloqueur selon un des deux axes de l'arbre quaternaire, jusqu'à ce que le rectangle résultant intersecte une arête du contour. Dans le cas où les axes de l'arbre quaternaire seraient légèrement désalignés avec l'agrégat de faces, le rectangle élargi pourrait "s'accrocher" sur une arête et ne pas atteindre sa pleine taille. Réduire le rectangle obtenu selon le premier axe (de typiquement 1/1000 de sa longueur) donne de meilleurs résultats en évitant ce problème⁶ (voir figure 3.6

⁵Initialement, aucune cellule intérieure n'est marquée.

⁶Une autre solution serait de détecter si l'accrochage se produit avec deux arêtes parallèles. Si c'est le cas, le système d'axes est légèrement désaligné et il peut être replacé correctement. On refait l'arbre quaternaire et on recommence l'extraction.

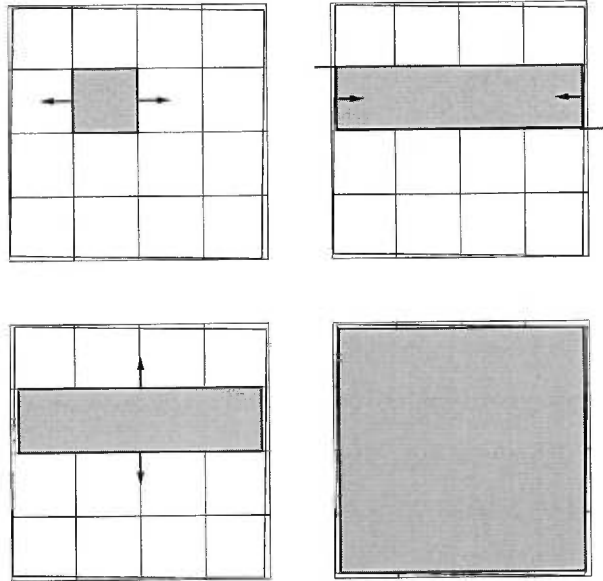


FIG. 3.6: Lors de l'élargissement du bloqueur, ce dernier s'accroche sur le contour (en haut à droite) et ne peut plus gonfler. Diminuer un peu le bloqueur permet d'améliorer la situation

pour exemple).

3. Élargir de façon similaire le bloqueur rectangulaire selon le second axe de l'arbre.
4. Marquer toutes les cellules de l'arbre quaternaire contenues à l'intérieur ou intersectant le bloqueur pour ainsi permettre qu'une nouvelle cellule non utilisée soit prise lors de la prochaine itération.
5. Itérer de nouveau à partir de l'étape 1, si jugé nécessaire.

Le processus arrête lorsque l'aire des éléments non marqués est plus petite qu'un seuil prédéterminé, ou qu'un nombre maximum de bloqueurs est atteint, ou encore qu'un nombre maximum d'itérations sans trouver de nouveaux bloqueurs s'effectue.

Les rectangles obtenus sont de simples polygones convexes qui peuvent se chevaucher entre eux. Ceci est un avantage voulu dans l'algorithme de visibilité puisque chaque bloqueur sera utilisé individuellement pour tester la visibilité entre deux agrégats quelconques. Par conséquent, chaque bloqueur se doit de couvrir le plus de superficie possible, même en chevauchant d'autres bloqueurs. Un autre avantage intéressant de cette technique d'extraction est que peu importe la polygonisation de la scène, les bloqueurs obtenus pour une même géométrie sont similaires. Les figures 3.7 et 3.8 montrent des exemples de bloqueurs extraits pour une scène de complexité moyenne. Dans toutes les scènes test (voir figure 4.13 page 64), l'extraction des bloqueurs et la formation de

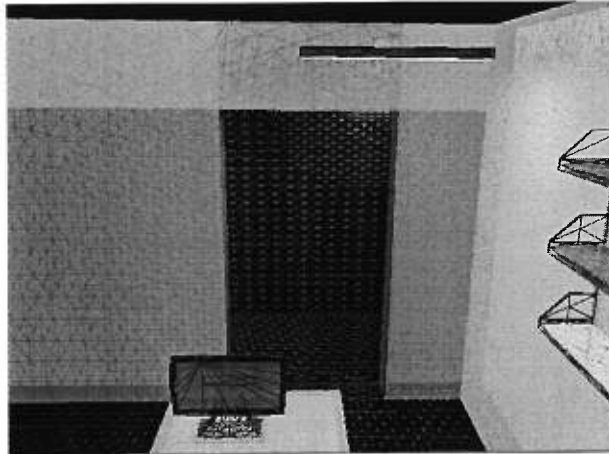


FIG. 3.7: Grands bloqueurs convexes extraits en semi-transparent apposés sur le maillage original de la scène

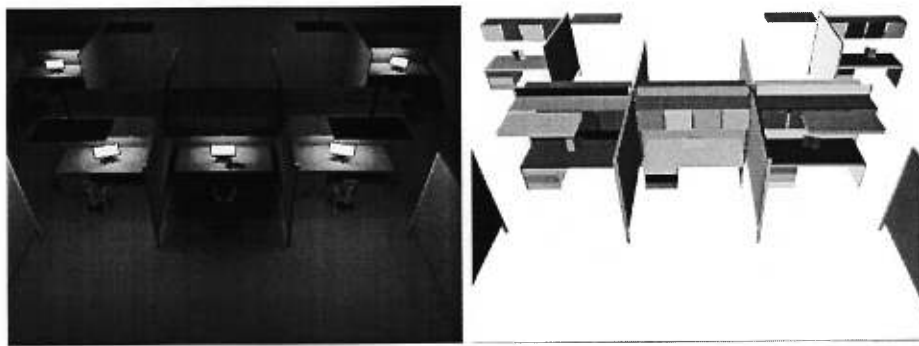


FIG. 3.8: La scène originale ainsi que les bloqueurs

la hiérarchie de la scène se sont effectuées en moins de 10 secondes sur un PC Athlon roulant Linux à 600 MHz.

Tous les bloqueurs extraits sont conservés dans une hiérarchie séparée de volumes englobants alignés sur les axes de la scène pour un accès rapide. L'algorithme 2 est utilisé une fois de plus afin de bâtir la hiérarchie de bloqueurs. Cette dernière est représentée sous la forme d'un tableau [Smi98] pour accéder de façon encore plus efficace aux bloqueurs en transformant la procédure d'accès récursive en méthode itérative.

3.2 Bloqueurs directionnels

Malgré que les bloqueurs planaires, introduits à la section précédente, permettent de prendre en considération une très grande portion du blocage dans une scène, ce type

Algorithme 3: Extraction de bloqueurs rectangulaires

```

Entrée : a : un agrégat de faces
Résultat : b : ensemble de bloqueurs

b ← ∅;
p ← Périmètre(a);
arbre ← ConstruireArbreQuaternaire(p);
tant que  $\sum Aire(\text{éléments internes}) > AireMin$  faire
    // trouver la plus grande cellule interne non marquée
    c ← GrandeCelluleInterne(arbre);
    // élargir le rectangle jusqu'à intersection avec le périmètre
    ÉlargirX(c, arbre);
    ÉlargirY(c, arbre);
    // marquer tous les cellules internes au rectangle
    MarquerCellules(c, arbre);
    b ← b ∪ {c};
fin
retourner b

```

de bloqueurs est inapproprié pour représenter certains autres cas d'occlusion. Ainsi, beaucoup de surfaces presque planaires, tel un mur de briques ou un dossier de chaise, peuvent générer de l'occlusion mais ne sont pas représentables par des bloqueurs planaires. D'autres types de surfaces peuvent aussi générer de l'occlusion malgré leur non planarité tels des draps ou des rideaux comme démontré à la figure 3.9. Ou encore, un groupe de bloqueurs planaires mais dont l'ensemble n'est pas planaire peut générer une occlusion plus importante sous certains angles, tel l'union de deux murs et du plafond et/ou du plancher. Il est alors essentiel d'introduire un nouveau type de bloqueurs pour tenir compte du phénomène d'occlusion de façon plus générale et complète.

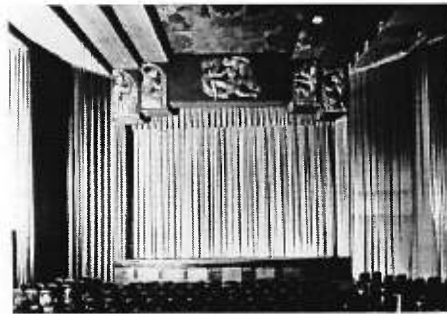


FIG. 3.9: Les rideaux coupent complètement la transmission de lumière entre la scène et la salle

Dans cette section est introduite la notion de bloqueur directionnel, se voulant une généralisation du concept de bloqueur planaire. Dans un premier temps, cette notion

est définie et par la suite un nouvel algorithme⁷ est introduit permettant d'extraire à partir d'une scène quelconque ce type de bloqueurs.

3.2.1 Définition

Un bloqueur directionnel est composé de deux entités. La première est la géométrie du bloqueur défini et construit de façon similaire aux bloqueurs planaires. La seconde entité est la *pyramide de visibilité* définie comme le demi-espace dans lequel le bloqueur est valide, c'est-à-dire que pour tout point contenu dans ce demi-espace, l'occlusion provoquée par le bloqueur directionnel est au plus aussi grande que la surface qu'il représente. Une illustration d'un bloqueur directionnel peut être observée à la figure 3.10.

Notre bloqueur directionnel est défini par un rectangle couvrant la plus grande superficie possible à l'arrière de la surface, et orienté, comme on le verra plus loin, selon son plan moyen afin d'obtenir la meilleure occlusion. Ainsi, notre bloqueur directionnel possède toutes les caractéristiques voulues, soit la simplicité⁸, une grande superficie, la convexité et il est conservateur. Cette dernière caractéristique est assurée par la pyramide de visibilité puisque son rôle consiste justement à définir une zone pour laquelle le bloqueur est conservateur.

Le demi-espace de la pyramide de visibilité est défini à l'aide de cinq plans : un plan pour restreindre chaque côté de la pyramide rectangulaire et s'appuyant sur les arêtes du bloqueur directionnel, et un dernier plan, pour la tronquer. Ce dernier plan, le plan de séparation, est situé tout juste au dessus de la surface, séparant le demi-espace de la surface. La base est inexistante et par conséquent la pyramide s'étend à l'infini.

3.2.2 Création

Identification des régions

Pour identifier les régions de la scène pouvant produire un bon niveau d'occlusion et pouvant former de bons bloqueurs directionnels, le même algorithme que dans le cas des bloqueurs planaires est utilisé. Par conséquent, la hiérarchie d'agrégats de faces

⁷Ce nouvel algorithme est aussi une extension de celui présenté à la section précédente pour extraire des bloqueurs planaires.

⁸Encore une fois, un rectangle est simple à tester et le test d'inclusion dans la pyramide de vue est efficace comme il est démontré au prochain chapitre.

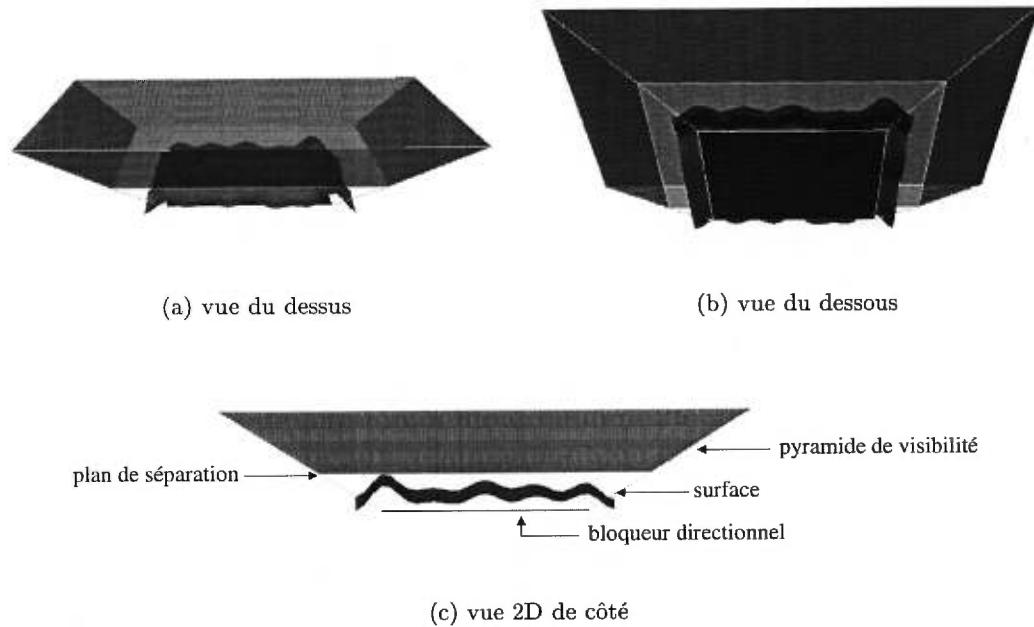


FIG. 3.10: Différentes vues d'un bloqueur directionnel

(section 3.1.1) doit être construite préalablement permettant de localiser aisément les grands agrégats de faces. La traversée de la hiérarchie est identique, mais le critère d'arrêt est changé pour obtenir des régions non planaires. On pourrait croire que remplacer le critère d'arrêt initial de planarité totale par un cône de normales ayant un angle maximal prédéfini est une bonne solution, mais malheureusement, ce n'est pas le cas. En effet, dans bien des situations, comme dans le cas d'un rideau, le cône des normales peut être tout près et même de 360° (une sphère) bien que cette surface puisse être un bon bloqueur.

Une solution serait d'essayer d'extraire pour chaque agrégat de faces un certain nombre de bloqueurs puis d'éliminer tous ceux considérés mauvais car étant trop petits ou ayant une utilité très restreinte (une étroite pyramide de visibilité). Le problème de cette solution est que le temps requis pour extraire tous ces bloqueurs et les analyser serait trop considérable, même pour une scène de complexité moyenne.

Notre solution à ce problème est basée sur l'épaisseur "relative" de la boîte englobante orientée des agrégats de faces. Cette épaisseur relative permet de calculer le degré de ressemblance entre la boîte et un plan, et est représentée comme le ratio entre la

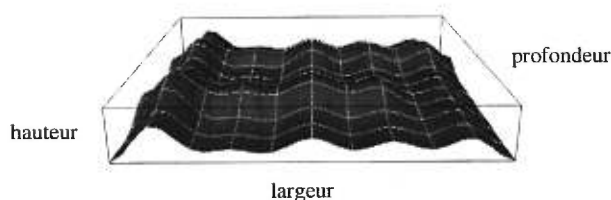


FIG. 3.11: Boîte englobante d'un agrégat

hauteur et la diagonale de sa base (voir figure 3.11) :

$$p = \frac{\text{hauteur}}{\sqrt{\text{largeur}^2 + \text{profondeur}^2}} \quad (3.13)$$

Une valeur proche de zéro indique qu'on est en présence d'une boîte plate et une grande valeur que celle-ci sort beaucoup du plan défini par sa base. Pour évaluer le potentiel d'un agrégat à produire de bons bloqueurs⁹, on fixe un seuil à l'épaisseur relative de la boîte englobante orientée. Un seuil de 0.2 offre généralement de bons résultats. Cependant quelques cas extrêmes ne sont pas reconnus par cette technique. Par exemple un rectangle avec une pointe en son milieu possède une mauvaise épaisseur relative selon notre critère mais peut provoquer beaucoup d'occlusion. Pour remédier à ce problème, on peut utiliser la boîte englobante orientée du contour de l'agrégat et non de l'agrégat au complet, mais alors encore d'autres problèmes pourraient se présenter.

Construction des bloqueurs

Après avoir déterminé les agrégats susceptibles de produire de bons bloqueurs directionnels, on doit les extraire puis évaluer leur qualité pour ne garder que les plus intéressants. Les différentes étapes requises pour construire un bloqueur directionnel sont illustrées par la figure 3.12. Comme première étape, on effectue la création du bloqueur de forme rectangulaire. Pour cela, on procède de la même façon que pour les bloqueurs planaires, mais cette fois, comme l'agrégat n'est pas nécessairement planaire, il faut déterminer le plan du bloqueur. Le plan choisi est en fait la base de la boîte englobante orientée de l'agrégat de faces. La boîte englobante orientée est calculée quant à elle lors de la construction de la hiérarchie de la scène. Pour obtenir son orientation,

⁹Il est important de rappeler que ce facteur permet d'évaluer la planarité d'un agrégat, mais on doit aussi tenir compte de la taille comme critère de sélection de bons agrégats et ceci est fait en traversant la hiérarchie à partir de la racine.

le vecteur propre ayant la plus grande valeur propre associée à l'agrégat correspondant est utilisé dans un premier temps. Puis les deux autres axes sont déterminés à l'aide des techniques décrites précédemment à la section 3.1.2. La projection orthographique de la surface de l'agrégat est utilisée pour extraire les bloqueurs à l'aide de l'arbre quaternaire qui est par la suite construit.

En second lieu, la pyramide de visibilité associée au bloqueur est construite. Pour ce faire, quatre plans perpendiculaires au bloqueur rectangulaire et appuyés sur ses côtés sont définis formant ainsi un prisme rectangulaire. Un dernier plan est ajouté, parallèlement au bloqueur au dessus de la boîte englobante, donc tout juste au dessus de l'extrémité de la surface de l'agrégat (voir figure 3.12c). Ce demi-espace ainsi défini est assurément conservateur en ce qui concerne l'occlusion du bloqueur puisque le prolongement de ce prisme en direction du bloqueur est entièrement contenu à l'intérieur du contour de l'agrégat.

Le seul problème est que ce demi-espace de visibilité est très restreint et par conséquent son utilisation demeurera très limitée. Pour améliorer son efficacité il faut augmenter le demi-espace de visibilité, ce qui implique la diminution du bloqueur. La dernière étape du processus de construction d'un bloqueur directionnel consiste dès lors au réajustement de l'espace de visibilité et du bloqueur associé dans le but d'obtenir le meilleur compromis entre la taille du bloqueur et celle du demi-espace.

Pour effectuer ce travail, on introduit une nouvelle métrique pour évaluer la qualité du bloqueur directionnel, et un algorithme permettant d'obtenir une bonne pyramide de visibilité associée à un bloqueur rectangulaire. Ainsi pour obtenir la solution désirée, on échantillonne l'espace des possibilités en diminuant successivement la taille du bloqueur directionnel, ajustant la pyramide puis évaluant sa valeur. Le bloqueur ayant la valeur la plus grande est gardé.

Puisque l'on veut qu'un bloqueur directionnel soit le plus grand possible et en même temps aussi que sa pyramide de visibilité, la métrique présentement utilisée consiste à multiplier l'aire du bloqueur par le volume d'une section parallèle à la pyramide. Cette section haute d'une unité débute au niveau du plan parallèle au bloqueur. La métrique est ainsi exprimée par l'équation suivante :

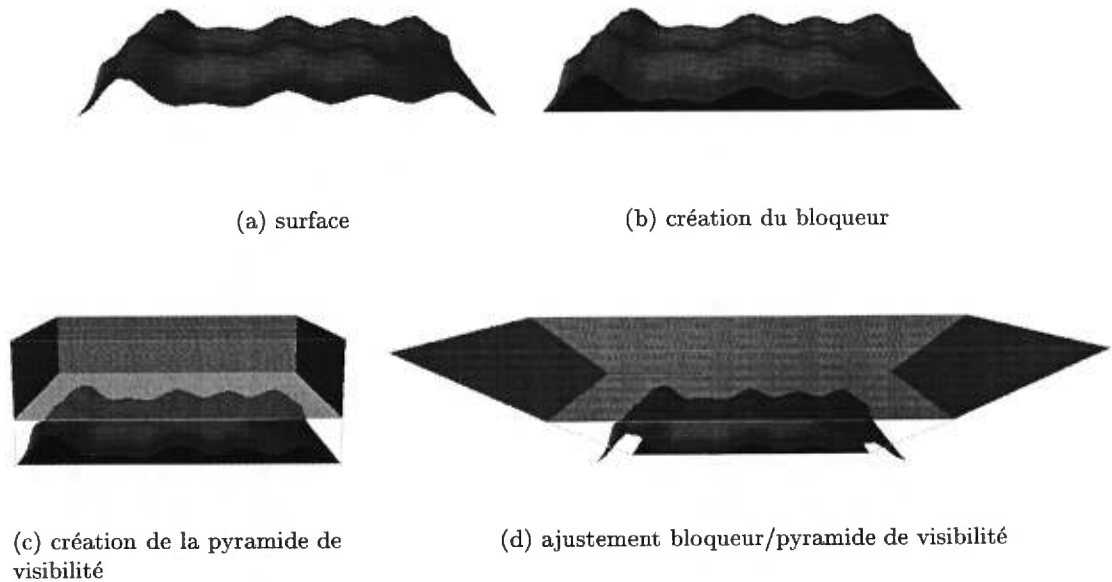


FIG. 3.12: Différentes étapes de la construction d'un bloqueur directionnel

$$C_b = (A_b)^\alpha \times V_p \quad (3.14)$$

où A_b est l'aire du bloqueur,
 V_p est le volume de la pyramide d'une unité de hauteur,
 α est le facteur d'amplification de l'aire.

Pour donner plus d'importance à l'aire du bloqueur pour obtenir des bloqueurs plus grands, l'aire est affectée, en exposant, d'un facteur d'amplification α . Une valeur de 3 donne de bons résultats. De plus, pour que cette métrique soit indépendante du facteur d'échelle de la scène, le bloqueur directionnel est transformé dans un espace canonique avant d'évaluer son coût. Pour ce faire, on n'a qu'à transformer le bloqueur directionnel pour que son aire initiale (avant ajustement de la pyramide) soit de un.

L'algorithme permettant de trouver une bonne pyramide de visibilité associée à un bloqueur est exprimé par l'algorithme 4. Le principe de cette méthode consiste à partir du prisme rectangulaire puis à élargir la base¹⁰ pour le transformer en pyramide. Cet élargissement s'arrête lorsqu'au moins une des arêtes du contour de l'agrégat touche à l'un de ses côtés. Pour y arriver, quatre listes d'arêtes sont formées contenant les

¹⁰Par base, on désigne le côté opposé sur lequel repose le bloqueur planaire (voir figure 3.12).

arêtes susceptibles d'entrer en contact avec chacun des côtés. Puis on évalue la distance minimale à laquelle le contact s'effectue pour chaque côté, en évaluant la distance de contact de chaque arête de la liste associée. Finalement, la distance minimale de tous les côtés est choisie et ainsi la pyramide est formée. La pyramide obtenue est proportionnelle au bloqueur directionnel et n'est pas nécessairement optimale. Une dernière étape pour améliorer la situation consiste à appuyer sur le contour tous les autres côtés de la pyramide qui ne sont pas en contact avec le contour de l'agrégat.

Algorithme 4: Ajustement de la pyramide de visibilité

```

Données : côté[i] : définition des côtés de la pyramide
// création d'une liste d'arêtes par côté
pour  $i \leftarrow 0$  à 3 faire
  | liste[i]  $\leftarrow \emptyset$ ;
  | pour chaque arête  $a$  du contour faire
  | | si  $a$  est au moins partiellement à l'extérieur du côté[i] alors liste[i]  $\leftarrow$  liste[i]  $\cup$   $a$ ;
  | fin
fin
// trouver la distance minimale de contact
distMin  $\leftarrow \infty$ ;
pour  $i \leftarrow 0$  à 3 faire
  | pour chaque arête  $a$  de liste[i] faire
  | | dist  $\leftarrow$  DistanceIntersection( $a$ , côté[i]);
  | | si  $dist < distMin$  alors distMin  $\leftarrow$  dist;
  | fin
fin
ajuster la pyramide selon DistMin;
ajuster les autres côtés;

```

3.3 Bloqueurs volumiques

Certains types d'objets peuvent être mieux représentés en terme d'occlusion par des bloqueurs volumiques que par des bloqueurs directionnels. Par exemple, un ballon peut utiliser plus efficacement la sphère intérieure comme bloqueur, ou encore un édifice peut être remplacé par un ou quelques prismes rectangulaires.

Andujar *et al.* [ASVN00] ont présenté un algorithme permettant de construire des bloqueurs volumiques représentant la scène sous forme d'un arbre octaïdre. Ils grossissent une première cellule intérieure à un objet jusqu'à atteindre les frontières de l'objet. Ensuite, ils recommencent avec une autre cellule intérieure, et ainsi de suite jusqu'à couvrir une bonne portion de l'intérieur de l'objet.

Les bloqueurs obtenus par cette technique possèdent les caractéristiques voulues

pour des bons bloqueurs et peuvent être utilisés en combinaison des bloqueurs directionnels dans divers algorithmes de visibilité, dont ceux présentés dans ce mémoire. Cependant, cette méthode requiert que les objets soient fermés pour ne pas provoquer d'erreurs de visibilité. Une façon de détecter automatiquement un bon nombre d'objets fermés entrant dans le cadre des méthodes présentées jusqu'à maintenant, est de traverser la hiérarchie d'agrégats de la scène et de sélectionner les agrégats ayant un périmètre nul. Un tel périmètre indique que toutes les arêtes de l'objet sont partagées par deux faces et que par conséquent l'objet est considéré fermé.

Les algorithmes de construction de bloqueurs présentés jusqu'à maintenant, produisent des bloqueurs convexes. Cette caractéristique est nécessaire dans plusieurs techniques [CT97, HMC⁺97, SVN99, LP00] afin de déterminer correctement l'occlusion provoquée par ces derniers sur la scène. Cependant, plusieurs autres techniques telles le tampon de profondeur hiérarchique [GK93] et les cartes d'occlusion hiérarchique [ZMHH97] n'ont pas cette exigence. C'est pourquoi, nous présentons dans la section suivante un nouvel algorithme permettant de produire de grands bloqueurs conservateurs, simples mais non nécessairement convexes.

3.3.1 Création de bloqueurs internes

La construction de ce type de bloqueurs est basée sur la technique de simplification par enveloppe de Cohen *et al.* [CVM⁺96]. On génère deux enveloppes entourant la surface du modèle géométrique, soit une interne et une externe. Ensuite, on peut utiliser un algorithme quelconque de simplification de maillages, toutefois en le contraignant à ce que le résultat de sa simplification demeure entre les deux enveloppes. Pour obtenir un bloqueur conservateur, on choisit le modèle original comme enveloppe externe et on fabrique une enveloppe interne au modèle. Un avantage de cette technique est que l'écart entre les deux enveloppes indique approximativement le pourcentage d'occlusion provoquée par le bloqueur en comparaison du modèle initial.

Création des enveloppes

La forme et le nombre de polygones du bloqueur résultant dépendent hautement de la forme des enveloppes. Comme mentionné précédemment, l'enveloppe externe est

simplement le modèle original pour assurer que le bloqueur produit soit complètement contenu à l'intérieur du modèle. Pour calculer l'enveloppe interne, on procède par rétrécissement du modèle en s'assurant que le résultat demeure valide (i.e. demeure à l'intérieur). Dans ce but, on calcule la normale à chaque sommet et on déplace chaque sommet de la distance souhaitée dans la direction inverse à sa normale (une normale pointe localement vers l'extérieur du modèle). On doit toutefois s'assurer que cette nouvelle enveloppe ne s'auto-intersecte pas. Pour ce faire, la technique décrite par Cohen *et al.* [CVM⁺96] est utilisée. Cette approche itérative déplace chaque sommet de l'enveloppe un par un par une fraction de la distance souhaitée. À chaque déplacement, les triangles connectés au sommet traité sont testés contre le reste de l'enveloppe pour s'assurer qu'il n'y ait pas d'intersection. S'il y a intersection, alors ce déplacement est rejeté. La figure 3.13 illustre un exemple d'enveloppe interne obtenue par ce procédé.

Le problème de cette technique pour calculer l'enveloppe interne est qu'il n'est pas évident a priori de choisir de quel pourcentage rétrécir le modèle original. Dans certaines configurations, lorsque l'enveloppe commence à s'auto-intersecter, des pointes apparaissent et limitent la simplification du modèle par la suite. Des techniques complémentaires peuvent être employées afin de résoudre ce problème.

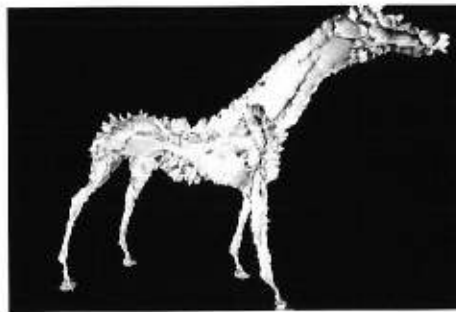


FIG. 3.13: Enveloppe interne

Simplification du modèle

De nombreuses techniques de simplification [Lue97, HG97, CMS98] peuvent être employées pour obtenir les bloqueurs. Nous avons implanté l'algorithme de Cohen *et al.* [CVM⁺96]. En premier lieu, une liste de tous les sommets du modèle original est formée; ces sommets seront éliminés un à un. Pour éliminer un sommet, tous les triangles y étant connectés sont détruits en même temps que ce dernier. Le trou ainsi formé est ensuite refermé avec de nouveaux triangles. S'il existe une configuration de

triangles permettant de refermer le trou sans intersecter les deux enveloppes, le sommet est éliminé et tous les sommets des nouveaux triangles sont ajoutés à la liste s'ils n'y étaient pas déjà. Si aucune configuration n'est satisfaisante, on revient au modèle précédent et on retire le sommet de la liste. L'algorithme s'arrête lorsque la liste est vide. Un exemple de bloqueur volumique utilisant cette méthode est présenté à la figure 3.14.

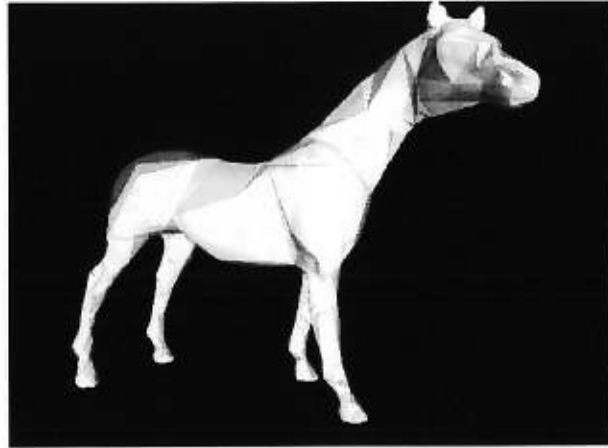


FIG. 3.14: Bloqueur volumique opaque (1482 triangles) et modèle original transparent (40000 triangles)

Pour être efficace, ces bloqueurs volumiques doivent être grands et contenir beaucoup moins de polygones que le modèle original. Il est donc important de s'assurer que le résultat obtenu par simplification est bon. De plus, les bloqueurs résultants ne sont pas convexes ce qui implique qu'ils ne peuvent être employés dans certains algorithmes.

Chapitre 4

Radiosité

Dans le chapitre précédent, un nouveau type de bloqueurs a été introduit, soit le bloqueur directionnel, de même que les processus de construction de ce dernier ainsi que de bloqueurs planaires et volumiques dans des scènes polygonales. Ces bloqueurs permettent de représenter l’occlusion des éléments d’une scène de façon conservatrice et à un coût moindre lorsque utilisés dans un algorithme d’élimination de surfaces cachées.

Ce chapitre se consacre à l’utilisation de tels bloqueurs dans un système de simulation de l’illumination globale par radiosité. Après avoir décrit ce qu’est un système de radiosité et les notions affiliées, un nouvel algorithme de visibilité sera introduit. Ce dernier utilise nos bloqueurs pour détecter les cas triviaux d’occlusion et de visibilité complètes. Son efficacité est par la suite démontrée sur des scènes affichant des caractéristiques variées.

4.1 Description de la radiosité

La radiosité est l’une des techniques les plus reconnues pour obtenir des images *photoréalistes*. Contrairement aux techniques basées sur le tracé de rayons [App68, Whi80] (abordées au prochain chapitre), la radiosité subdivise une scène en éléments et résout le système simplifié d’équations ainsi formé. Ceci permet d’obtenir des effets de diffusion des couleurs dus à l’éclairage indirect, caractéristique importante pour simuler un éclairage physiquement correct, et donc, réaliste.

Cette technique empruntée de l’ingénierie thermique [HS67] a été introduite récemment, au milieu des années 1980, en image de synthèse par Goral *et al.*

[GTGB84]. Toutefois, les premières formulations remontent au début du siècle [Yam26]. Mais, ce n'est qu'un an après son apparition en image de synthèse que la visibilité fut tenue en compte, par la méthode d'hémi-cubes [CG85]. Dès le début, cette méthode fut restreinte aux échanges effectués entre surfaces parfaitement diffuses, c'est-à-dire réfléchissant également en intensité dans toutes les directions. Certaines extensions ont permis d'étendre la simulation aux surfaces parfaitement spéculaires [ICG86, RT90] et aux surfaces de type *lustré* [AH93], mais encore aujourd'hui ces techniques sont peu employées à cause de leurs coûts importants en mémoire et en temps de calcul. Il est à noter cependant, qu'avec la venue d'un algorithme présenté par Granier *et al.* [GDW00], il est désormais possible d'étendre la radiosité pour tenir compte de tous les types de surfaces (*BRDF* quelconque), ceci en combinant les éléments finis avec le tracé de rayons stochastique.

La présente section est divisée en deux parties. La première décrit de façon générale la radiosité ainsi que ses différentes techniques de résolution. La seconde décrit plus en détail l'algorithme de radiosité qui est utilisé en conjonction avec notre méthode de visibilité.

4.1.1 Survol de la radiosité

Cette section se veut un bref synopsis du domaine de la radiosité. Pour de plus amples précisions sur le sujet, les lecteurs sont conviés à consulter les livres de Sillion et Puech [SP94] et de Cohen et Wallace [CW93b] consacrés à la radiosité.

L'illumination globale est un phénomène fort complexe dans lequel les photons émis par des sources lumineuses interagissent un très grand nombre de fois avec les surfaces de la scène. Kajiya a modélisé la généralité de ce phénomène sous la forme de l'équation du rendu [Kaj86] :

$$L(x, x') = g(x, x') \left[\epsilon(x, x') + \int_S \rho(x, x', x'') L(x', x'') dx'' \right] \quad (4.1)$$

où x, x', x'' sont des points appartenant aux surfaces,
 S représente les surfaces de la scène,
 $L(x, x')$ est la radiance allant de x' à x ,
 $g(x, x')$ vaut 0 si le point x ne voit pas x' , et
vaut $\frac{1}{r^2}$ sinon (r est la distance entre x et x'),
 $\epsilon(x, x')$ est l'intensité lumineuse émise de x' vers x ,
 $\rho(x, x', x'')$ est la BRDF au point x' , pour la lumière partant de x''
passant par x' et allant vers x .

Cette équation représente l'énergie lumineuse arrivant en un point d'une surface en provenance d'un autre point. L'énergie de ce second point est définie à son tour par son émission en direction du premier et de l'ensemble des émissions et réflexions de tous les points des surfaces de la scène en entier vers celui-ci. Ainsi, ces dernières émissions sont définies de façon récursive en réutilisant la même équation. Le résultat de cette équation, la radiance en un point, se définit comme le flux d'énergie associé à un pinceau d'angle solide $d\omega$ et traversant une surface d'aire perpendiculaire à la direction de propagation. Par conséquent, les unités sont les $\frac{Watt}{m^2 sr}$.

Ce modèle général tient compte de tous les types d'interactions possibles entre la lumière et les surfaces et est noté en expression régulière de la façon suivante : $\mathcal{L}(\mathcal{S}|\mathcal{D})^*\mathcal{E}$ où \mathcal{L} est la lumière, \mathcal{S} une interaction avec une surface spéculaire, \mathcal{D} une interaction avec une surface diffuse et \mathcal{E} une interaction arrivant à l'oeil. Résoudre un tel système s'avère être une tâche très coûteuse et par surcroît, les solutions proposées à ce jour [CPC84, Kaj86, LW93, Jen96, VG97] demeurent souvent bruitées. En se restreignant exclusivement aux échanges effectués entre des surfaces diffuses ($\mathcal{L}\mathcal{D}^*\mathcal{E}$), il est possible d'obtenir une solution plus continue en un temps plus raisonnable, qui est par ailleurs indépendante du point de vue de l'observateur. En imposant ces nouvelles restrictions à l'équation du rendu et en la reformulant de telle manière à obtenir la radiosité en un point (intégrer l'équation de rendu selon l'hémisphère du point au-dessus de sa surface), on obtient l'équation générale de la radiosité (équation 4.2).

$$B(x) = E(x) + \rho(x) \int_S B(x') G(x, x') dA'. \quad (4.2)$$

Les techniques de radiosité se proposent de résoudre ce système d'équations ne traitant que les surfaces diffuses, aussi appelées surfaces lambertiennes, en discrétisant la scène en éléments tel que démontré par la figure 4.1. En découpant les surfaces de la scène en n éléments, et en supposant que la solution est constante sur toute la surface d'un élément, on transforme l'équation générale de la radiosité (équation 4.2) sous la forme discrète suivante :

$$B_i = E_i + \rho_i \sum_{j=1}^n B_j F_{ij} \frac{A_j}{A_i} \quad (4.3)$$

où B_i est la radiosité de l'élément i ,
 E_i est l'émission de l'élément i ,
 ρ_i est la réflectance de l'élément i ,
 F_{ij} est le *facteur de forme* entre les éléments i et j ,
 A_i est l'aire de l'élément i .

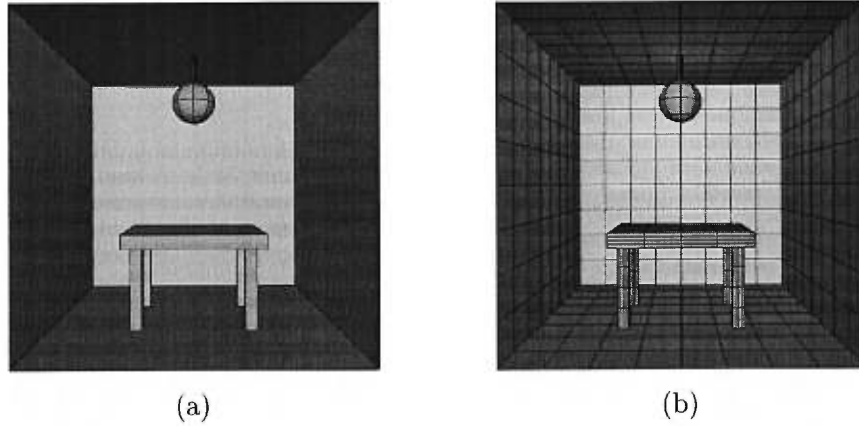


FIG. 4.1: a) scène originale ; b) scène après discrétisation en éléments

Sous cette forme, on obtient un système d'équations linéaires ayant pour solution la radiosité de chacun des éléments. Ainsi, la radiosité d'un élément se définit comme l'émission de l'énergie lumineuse de l'élément plus la somme de la radiosité de tous les éléments visibles de celui-ci et chaque élément est pondéré par sa réflectance ainsi que par la fraction de visibilité appelée le facteur de forme.

En fait, en résolvant ce système d'équations, les techniques de radiosité vont tenter de reconstruire une fonction scalaire située sur la surface, représentant l'intensité lumineuse (la radiosité) pour chaque point de toutes les surfaces de la scène. Sous la

forme présentée par l'équation 4.3, la résolution et la reconstruction de cette fonction de hauteur sur la surface sera effectuée à partir d'une base constante. Cependant, n'importe quel type de base peut être utilisé. Par exemple des bases linéaires, quadratiques et même des ondelettes ont déjà été utilisées, mais néanmoins les bases constantes demeurent les plus utilisées pour leur simplicité et leur moindre coût¹. La figure 4.2 est un exemple démontrant la différence entre la fonction originale à reproduire et l'approximation faite à l'aide de différentes bases.

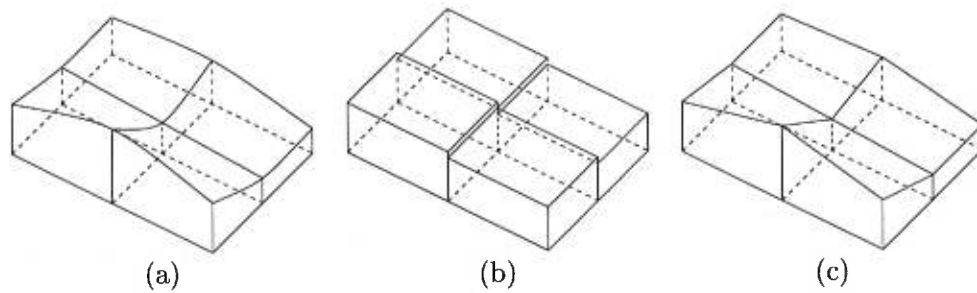


FIG. 4.2: a) fonction originale; b) base constante; c) base linéaire

Pour résoudre le système d'illumination globale, les techniques de radiosité classiques² divisent en six étapes (figure 4.3) le processus allant de la modélisation de la scène à l'affichage du résultat dans une image à partir d'un point de vue spécifié. Après avoir modélisé la scène, il faut bien la discrétiser en éléments. Le résultat de la simulation dépend directement mais non exclusivement de cette discrétisation. Il est donc important de choisir la bonne taille et la bonne forme pour reproduire tous les effets importants de l'illumination afin d'obtenir une solution juste et précise selon les besoins. Par la suite, les facteurs de forme entre chaque paire d'éléments (au nombre de n^2 où n est le nombre d'éléments) sont calculés. Cette étape comprenant la détermination de la visibilité entre chaque paire d'éléments domine généralement le coût de tout le processus de la radiosité. Il est à noter cependant que cette étape sera incorporée dans le processus de résolution du système d'équations linéaires dans les techniques plus avancées.

À partir des coefficients (facteurs de forme et réflectances) et des équations posées,

¹Plus la base est simple moins il y a de coefficients à calculer.

²On verra un peu plus loin que les techniques hiérarchiques modifient un peu l'ordre et le cheminement du processus.

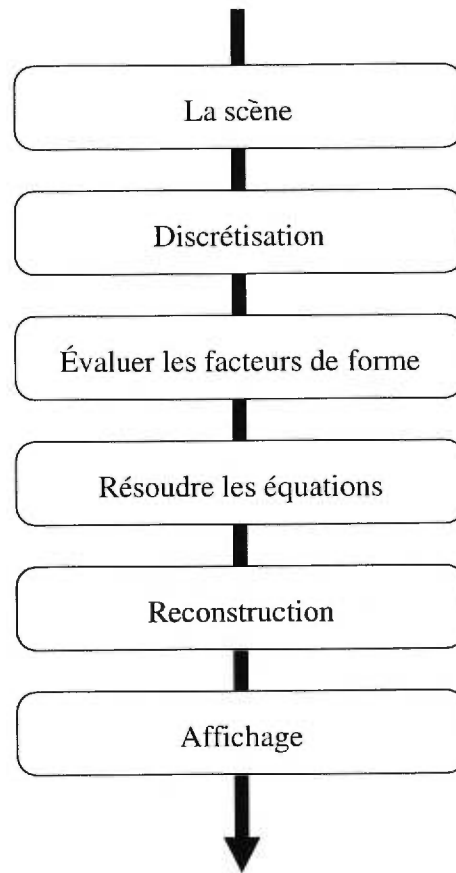


FIG. 4.3:  tapes du processus de r solution par radiosit 

la solution de radiosit  de chaque  l ment est trouv e lors de l' tape centrale du processus. Diff rentes m thodes de r solution sont d crites un peu plus loin dans cette section. La derni re  tape de ce long processus avant le rendu de la solution consiste   reconstruire la fonction continue de radiosit  sur les surfaces. Cette  tape est bien entendu, d pendante de la base utilis e lors des processus de discr tisation et de r solution.

Le maillage

Les  l ments utilis s lors du calcul de radiosit  se doivent d' tre simples pour permettre l' valuation correcte et rapide de leur base. C'est pourquoi en g n ral les  l ments consistent exclusivement en un ensemble de rectangles et de triangles. De plus, il est pr f rable dans bien des cas³ que la forme des  l ments soit le plus compact

³Lorsque la distribution g n rale de l'illumination n'est pas connue il est pr f rable de garder une densit  uniforme mais dans le cas contraire si les discontinuit s ont  t  pr alablement calcul es [LTG93], un maillage non uniforme est plus adapt .

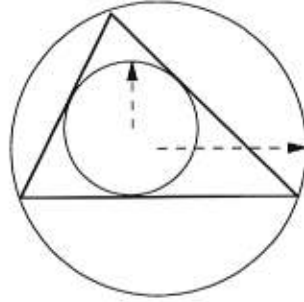


FIG. 4.4: Calcul du rapport hauteur/largeur d'un triangle

possible. La compacité d'un élément peut être obtenue en tentant d'augmenter son rapport hauteur/largeur qui se définit comme le ratio entre le rayon du cercle englobant la primitive et celui étant contenu dans cette dernière. Un exemple est donné pour un triangle à la figure 4.4.

Le niveau de subdivision des éléments dépend de la qualité et de la précision de la solution désirée et influence directement le temps de calcul de cette solution. Lors de la reconstruction, il faut s'assurer d'éviter les sommets en T, c'est-à-dire un sommet n'arrivant sur une arête que d'un élément, et donc non partagé par chacun des éléments adjacents (voir figure 4.5). Ce type de sommets va créer des discontinuités fictives (non voulues) dans la solution s'ils ne sont pas éliminés avant de reconstruire la solution finale de radiosité. Ce problème peut être résolu lors de la discrétisation de la scène, mais certains préfèrent attendre à l'étape de reconstruction diminuant ainsi quelque peu le nombre d'éléments requis pour résoudre le système. Pour solutionner le problème, il suffit de découper l'élément voisin au sommet en T en créant de nouvelles arêtes joignant ce sommet aux autres sommets de l'élément fautif. Pour obtenir une meilleure subdivision lors de la résolution de ce problème, différentes techniques ont été proposées [BMSW91, She88].

Facteur de forme

Le facteur de forme représente la relation géométrique entre deux éléments ou surfaces. À l'aide de ce facteur, on peut évaluer la quantité d'énergie lumineuse qui sera transmise d'une surface à une autre.

On définit le facteur de forme entre deux surfaces différentielles par l'équation 4.4 où θ_i représente l'angle entre la normale à la surface différentielle di et le vecteur allant de di à dj , r est la distance entre les deux surfaces différentielles et V est le

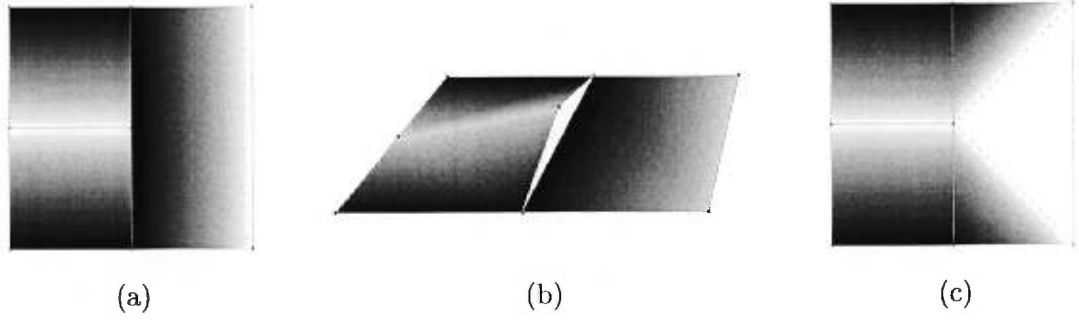


FIG. 4.5: a) Sommet en T ; b) vue en 3D ; c) une solution au problème

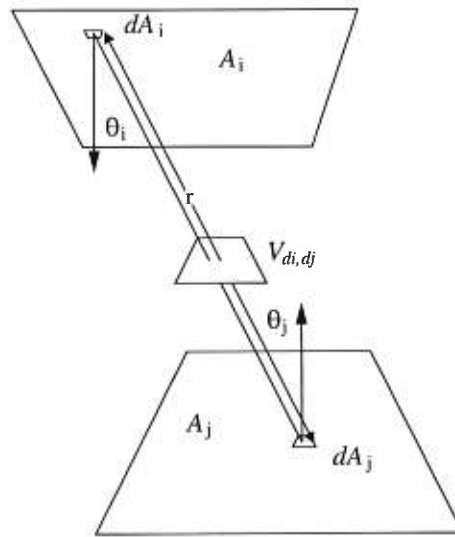


FIG. 4.6: Facteur de forme et ses composantes

facteur de visibilité (0 ou 1 selon que les surfaces différentielles sont visibles entre elles).

$$dF_{di,dj} = \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{di,dj} dA_j. \quad (4.4)$$

Pour calculer le facteur de forme entre une surface différentielle et un élément, il faut intégrer le facteur de forme précédent (entre deux surfaces différentielles) sur toute la surface de l'élément.

$$F_{di,j} = \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{di,dj} dA_j. \quad (4.5)$$

Enfin, pour obtenir le facteur de forme désiré entre deux éléments, pour résoudre le système d'équations linéaires de la radiosité, il faut intégrer sur la surface des deux éléments.

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{di,dj} dA_j dA_i. \quad (4.6)$$

Pour résoudre cette équation, deux techniques peuvent être utilisées : une solution analytique et une solution numérique. Différentes configurations précises existent sous forme analytique comme entre une surface différentielle et un polygone [BRW89], deux polygones [SH93], une surface différentielle et un disque. Pour toutes les configurations, une solution par échantillonnage (numérique) [CG85, Pue91] peut être utilisée. Lorsque l'on veut évaluer le facteur de forme entre deux petits éléments distants, il est commun d'approximer le résultat en calculant le facteur de forme entre une surface différentielle et un polygone ou même un disque. L'équation suivante (4.7) permet de calculer le facteur de forme entre une surface différentielle et un polygone sans tenir compte de la visibilité. Cette dernière doit être calculée séparément, que cela soit par lancer de rayons (échantillonnage) ou bien en découpant les éléments entre eux pour ne garder que les parties visibles.

$$F_{di,j} = \frac{1}{2\pi} \sum_{k=1}^n \beta_k \cos \alpha_k. \quad (4.7)$$

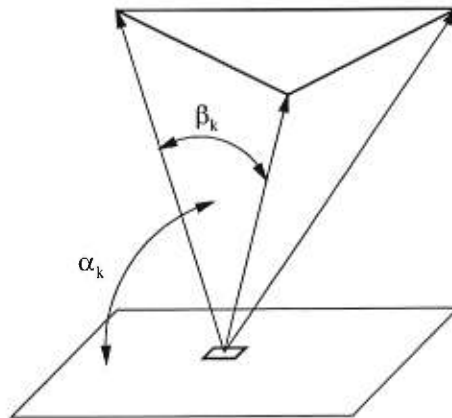


FIG. 4.7: Facteur de forme entre une surface différentielle et un polygone de n côtés

Résolution du système d'équations

Dans la littérature mathématique, il existe plusieurs techniques permettant de résoudre un système d'équations linéaires. Basées sur ces dernières, différentes solutions ont été proposées jusqu'à ce jour pour résoudre le système spécifique de la radiosité. L'équation discrète de la radiosité (équation 4.3) peut être formulée sous forme matricielle (équations 4.8 et 4.9) où \mathbf{E} est le vecteur des émissions, \mathbf{B} le vecteur des radiosités (les inconnues) et \mathbf{K} la matrice des facteurs de forme (avec les réflectances).

$$\mathbf{KB} = \mathbf{E} \quad (4.8)$$

$$\begin{bmatrix} 1 - \rho_1 F_{1,1} & \cdots & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & \cdots & -\rho_2 F_{2,n} \\ \vdots & & \vdots \\ -\rho_{n-1} F_{n-1,1} & \cdots & -\rho_{n-1} F_{n-1,n} \\ -\rho_n F_{n,1} & \cdots & 1 - \rho_n F_{n,n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{n-1} \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_{n-1} \\ E_n \end{bmatrix}. \quad (4.9)$$

Une méthode directe telle que l'élimination de Gauss [GTGB84] peut être utilisée pour solutionner le système. Cependant cette technique exige de garder en mémoire tous les facteurs de forme ($O(n^2)$ où n est le nombre d'éléments) et le temps de solution est de $O(n^3)$. D'autres techniques, itératives, plus appropriées et moins gourmandes en mémoire permettent d'obtenir un ensemble de solutions intermédiaires. On dénombre parmi celles-ci les techniques de relaxation de Jacobi, Gauss-Seidel et Southwell. La méthode de Gauss-Seidel se veut une méthode d'accumulation tandis que celle de Southwell a pour représentation géométrique la propagation de l'énergie lumineuse dans la scène. En effet, lors d'une étape de Gauss-Seidel, un élément accumule l'énergie de tous les autres éléments de la scène et après une itération complète, chaque élément a reçu l'énergie de tous les autres. De nombreuses itérations sont requises pour converger vers la solution.

Une variante de Southwell introduite par Cohen *et al.* [CCWG88] est préférée aux techniques précédentes car elle converge beaucoup plus rapidement vers une solution visuellement satisfaisante. Dans cette méthode, à chaque étape on choisit l'élément ayant le plus d'énergie lumineuse à propager, puis on propage cette énergie à tous les

autres éléments de la scène. Chaque élément absorbe une quantité d'énergie et met à jour sa propre quantité d'énergie à propager. De cette façon, lors des premières étapes, les lumières contenues dans la scène émettent leur énergie au reste des éléments, donnant plus rapidement une solution intéressante et près de la solution finale. Le pseudo-code de cette méthode est présenté à l'algorithme 5.

Algorithme 5: Radiosité par propagation

```

Données :  $B$  : la radiosité des éléments
            $\Delta B$  : la radiosité à propager des éléments
Résultat :  $B$ 

pour chaque élément  $i$  faire
     $B_i \leftarrow E_i$ ;
     $\Delta B_i \leftarrow E_i$ ;
fin
tant que précision voulue non atteinte faire
    Sélection d'un élément  $i$  tel que  $\Delta B_i$  est le plus grand;
    pour chaque élément  $j$  faire
        // Propagation de  $\Delta B_i$  à l'élément  $j$  et  $j \neq i$ 
         $\Delta \text{Radiosité} \leftarrow \rho_j \Delta B_i F_{ij} \frac{A_i}{A_j}$ ;
         $\Delta B_j \leftarrow \Delta B_j + \Delta \text{Radiosité}$ ;
         $B_j \leftarrow B_j + \Delta \text{Radiosité}$ ;
    fin
     $\Delta B_i \leftarrow 0$ ;
fin
  
```

Même si ces techniques requièrent moins de mémoire ($O(n)$) le temps de résolution est au moins de $O(n^2)$ sans compter le coût du calcul des facteurs de forme qui est au mieux $O(\log n)$ par élément à cause du facteur de visibilité. Pour accélérer la vitesse de résolution du système, des méthodes hiérarchiques à deux niveaux [CGIB86] puis à multi-niveaux [HSA91] ont été proposées. Ces techniques se basent sur le fait que l'échange d'énergie entre deux régions éloignées peut s'approximer en un seul échange au lieu d'un ensemble d'échanges entre chacun des éléments des groupes. Ceci a pour résultat de diminuer le nombre d'échanges à effectuer de $O(n^2)$ à $O(n)$. La complexité du problème fut même réduite à $O(\log n)$ [SAG94, Sil94] en regroupant par agrégats les éléments de la scène.

Reconstruction de la solution de radiosité

Une fois que la solution de radiosité a été calculée, il peut être nécessaire (voir figure 4.8), surtout si les bases utilisées sont constantes, de reconstruire une nouvelle fonction de hauteur pour représenter la radiosité pour chaque point des surfaces. Dans

ce dernier cas, une solution simple consiste à calculer la valeur de radiosité aux sommets des éléments à partir des valeurs des éléments joints à ces sommets. Différents arrangements existent pour calculer une moyenne des valeurs de radiosité selon la taille et la connexité des éléments. La valeur d'un point contenu sur un élément peut ensuite être obtenue en effectuant une interpolation bilinéaire. Une approximation plus rapide car implantée au niveau du matériel est d'effectuer une interpolation de Gouraud [Gou71] en espace écran. Ces techniques permettent d'obtenir une continuité C^0 qui peut être discernée aisément dans certaines configurations. Alors on peut utiliser d'autres techniques [Max89, Rei92] pour obtenir une meilleure continuité C^1 et plus encore.

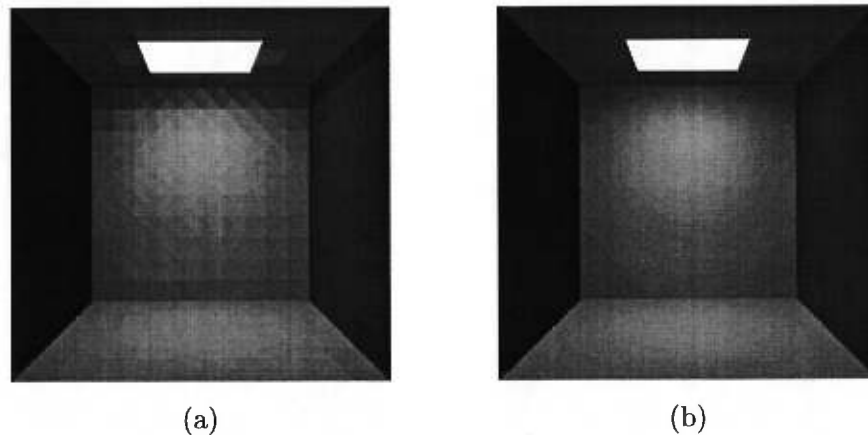


FIG. 4.8: a) base constante ; b) reconstruction par interpolation Gouraud

4.1.2 Techniques implantées

Bien que notre nouvel algorithme de visibilité (présenté plus loin) peut être appliqué à tous les types de systèmes de résolution de la radiosité, il a spécialement été adapté à un système hiérarchique par propagation avec agrégation. Pour pouvoir bien comprendre l'algorithme de visibilité ainsi que mettre en perspective les résultats obtenus, les spécificités de l'algorithme de radiosité utilisé sont décrites dans la section qui suit.

Radiosité hiérarchique par propagation

L'algorithme de radiosité implanté s'inspire de celui introduit par Stamminger *et al.* [SSSS98] et dont le pseudo-code est présenté par les algorithmes 6 à 9. Chacune

des itérations de l'algorithme correspond à un niveau d'indirection de la lumière. Conséquemment lors de la première itération, l'éclairage direct est calculé, puis lors de la seconde, un premier niveau d'indirection de l'illumination est ajouté à la scène, et ainsi de suite.

Le travail effectué lors d'une itération se divise en trois parties : la propagation à différents niveaux, la redistribution dans la hiérarchie de la radiosité propagée et la mise à jour des radiosités accumulées et à propager. Dans un premier temps l'énergie lumineuse est propagée de la scène à toute la scène. Pour ce faire, à chaque échange potentiel d'énergie entre un émetteur et un receveur (initialement la scène à la scène) une procédure, nommée l'*oracle*, choisit si l'échange doit avoir lieu à ce niveau ou si l'on doit plutôt subdiviser soit l'émetteur ou soit le receveur. L'*oracle* utilisé est basé sur le critère de raffinement *BFA* [HSA91, SAG94]. Ce dernier consiste à approximer l'erreur produite en effectuant l'échange à ce niveau en pondérant la radiosité à émettre par le facteur de forme entre l'émetteur et le receveur (sans visibilité), et par l'aire du receveur.

Lorsque l'échange se produit, on calcule le facteur de forme entre les deux éléments (incluant la visibilité), puis on transmet l'énergie lumineuse à l'élément receveur. Après que tous les éléments aient propagé leur énergie lumineuse, celle-ci se retrouve distribuée à différents niveaux de la hiérarchie. Pour trouver l'énergie totale de chaque élément, il faut la redistribuer jusqu'aux feuilles (en sommant l'énergie) puis on la remonte en la distribuant aux parents. Le tout est décrit par l'algorithme 8. Par la suite, les radiosités accumulées et à propager des éléments sont mises à jour pour l'ensemble de la scène et une nouvelle itération peut être commencée à nouveau si jugé nécessaire.

La hiérarchie employée pour regrouper tous les éléments en une racine commune est une hiérarchie d'agrégats de faces telle que décrite lors du chapitre précédent.

Facteur de forme

Puisque la plupart des échanges s'effectuent entre agrégats de faces ou entre un agrégat et un élément, on ne peut utiliser directement de formule analytique (aucune formule n'existe pour une forme quelconque). On se voit alors contraint d'approximer le facteur de forme par échantillonnage. La procédure adoptée est de choisir un certain

Algorithme 6: Système de résolution de radiosité hiérarchique par propagation

Données : \hat{B} : radiosité des éléments
 $\Delta\hat{B}$: radiosité à propager
 $\Delta\hat{B}'$: radiosité à propager prochainement

Résultat : \hat{B}

$\hat{B} \leftarrow 0;$
 $\Delta\hat{B}' \leftarrow 0;$
 $\Delta\hat{B} \leftarrow \hat{E};$
tant que *précision voulue non atteinte* **faire**
 TransporteLumière(racine, racine);
 PushPull(racine, 0);
 $\hat{B} \leftarrow \hat{B} + \Delta\hat{B}';$
 $\Delta\hat{B} \leftarrow \Delta\hat{B}';$
 $\Delta\hat{B}' \leftarrow 0;$
fin

Algorithme 7: TransporteLumière(p, q)

Entrée : p : élément émetteur
q : élément receveur

Données : $\Delta\hat{B}'_p$: radiosité de l'élément p à propager prochainement
 $\Delta\hat{B}'_q$: radiosité de l'élément q à propager
 \hat{T}_{pq} : facteur de forme entre p et q \times réflectance de q

suivant *Oracle(p, q)* **faire**
 cas où égale p
 pour chaque enfant c de p faire TransporteLumière(c, q);
 cas où égale q
 pour chaque enfant c de q faire TransporteLumière(p, c);
 autres cas
 $\Delta\hat{B}'_p \leftarrow \Delta\hat{B}'_p + \hat{T}_{pq} \Delta\hat{B}'_q;$
fin

Algorithme 8: PushPull(p, \hat{B}_{bas})

Entrée : p : un élément
 \hat{B}_{bas} : radiosité accumulée par les parents de p

Données : $\Delta\hat{B}'_p$: radiosité de l'élément p à propager prochainement

si p *est une feuille* **alors**
 $\hat{B}_{haut} \leftarrow \Delta\hat{B}'_p + \hat{B}_{bas};$
sinon
 $\hat{B}_{haut} \leftarrow 0;$
 pour chaque enfant c de p faire
 $\hat{B}_{tmp} \leftarrow PushPull(c, \Delta\hat{B}'_p + \hat{B}_{bas});$
 $\hat{B}_{haut} \leftarrow \hat{B}_{haut} + \hat{B}_{tmp} \frac{Aire(c)}{Aire(p)};$
 fin
 $\Delta\hat{B}'_p \leftarrow \hat{B}_{haut};$
 retourner \hat{B}_{haut}
fin

Algorithme 9: Oracle(p, q)

```

Entrée : p : élément émetteur
         q : élément receveur
Données : seuil : erreur minimum que l'on veut permettre
         tailleMinimum : taille minimum que peut avoir un élément

erreur ← CalculErreur();
si erreur > seuil alors
    si Taille(p) > tailleMinimum alors
        Subdivise(p);
        retourner p;
    sinon
        Subdivise(q);
        retourner q;
    fin
sinon
    retourner aucun;
fin

```

nombre de points ⁴ distribués uniformément sur la surface de l'élément ou de l'agrégat selon le cas, et de calculer le facteur de forme entre chaque point et l'autre surface en utilisant la formule analytique d'un point à un disque :

$$F_{ij} = \frac{A_j \cos \theta_i \cos \theta_j}{\pi r^2 + A_j} \quad (4.10)$$

où A_j est l'aire du disque,
 θ_i est l'angle entre la normale au point i et le vecteur allant de i au centre du disque,
 θ_j est l'angle entre la normale du disque et le vecteur allant du centre du disque à i ,
 r est la distance du point au centre du disque.

Pour que l'approximation soit bonne, il est important que les agrégats soient le plus compact possible et aient une forme s'approchant de celle d'un disque comme décrit au chapitre précédent. La visibilité entre les deux surfaces est calculée dans une étape séparée par lancer de rayons. Un nombre n d'échantillons distribués uniformément est choisi sur la première surface. Pour chacun d'entre eux, un point est choisi sur la deuxième surface (encore une fois distribués uniformément), puis un rayon est lancé pour tester la visibilité. Le résultat du test est binaire pour un rayon mais pour la surface la visibilité est le pourcentage de rayons ayant atteint leur cible.

⁴8 échantillons semblent être un bon compromis entre la qualité et le temps de calcul.

Pour générer un ensemble de points distribués uniformément sur la surface d'un triangle, la technique de Turk [Tur90] est employée. Cette technique utilise les coordonnées barycentriques pour parvenir à ses fins. Dans un premier temps, deux des trois coordonnées sont calculées en générant deux nombres aléatoires entre $[0, 1]$, puis si leur somme est supérieure à un, on recalcule chaque valeur en les soustrayant à un. Finalement la troisième coordonnée est calculée directement à l'aide des deux premières. Cette technique est étendue à un agrégat de faces simplement en descendant sa hiérarchie jusqu'à atteindre une de ses feuilles (un triangle). Lors de la descente de cette hiérarchie binaire, le prochain noeud est choisi de la façon suivante : un nombre aléatoire est généré entre $[0, 1]$ et si ce nombre est compris entre $[0, \alpha]$ (où α est l'aire du noeud gauche sur l'aire du noeud courant) le noeud de gauche est choisi sinon le noeud droit est choisi.

Subdivision du maillage

Lorsque l'oracle décide de subdiviser une surface, s'il s'agit d'un agrégat de volumes ou de faces ou encore d'un élément subdivisé préalablement, aucun travail n'est requis. Cependant, si l'élément en question est une feuille de la hiérarchie, il doit être subdivisé. Dans notre implantation, on est en possession d'une hiérarchie binaire de triangles, par conséquent chaque élément à subdiviser est partagé en deux triangles. De plus, on veut éviter de produire des sommets en T pour les raisons mentionnées préalablement.

Notre solution consiste à diviser le triangle à partir du point milieu de l'arête la plus longue et du sommet opposé, ceci permettant de garder une forme assez compacte. Puis, pour éliminer le sommet en T tout nouvellement créé, le triangle opposé est divisé à son tour. Si l'arête partagée des deux triangles et contenant le sommet en T est aussi la plus longue du second triangle, alors il suffit de subdiviser le triangle de la même façon que le premier et le processus est terminé. Toutefois, dans le cas contraire, le second triangle est subdivisé selon son arête la plus longue (formant un nouveau sommet en T) et ensuite les deux sommets en T sont reliés ensemble (voir figure 4.9 pour un exemple visuel). Finalement de façon récursive, le triangle opposé au nouveau sommet en T est subdivisé à son tour.

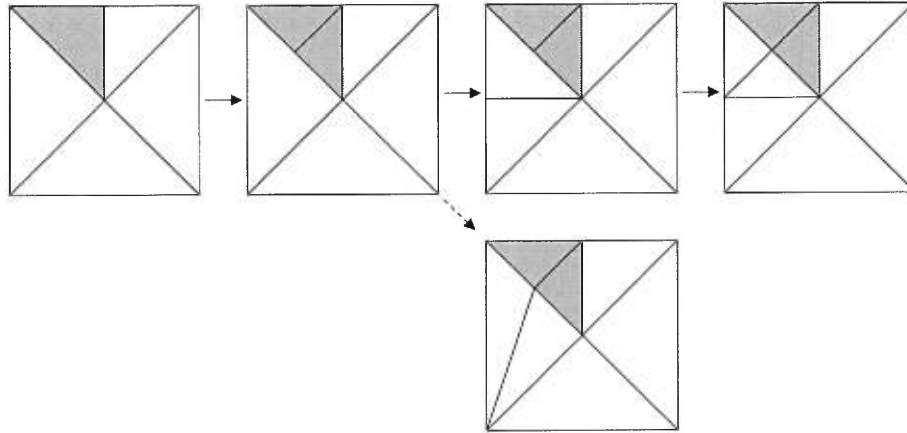


FIG. 4.9: Étapes du processus de subdivision d'un triangle (les flèches pleines représentent les bonnes étapes et la flèche pointillée représente une mauvaise subdivision provoquant des triangles allongés)

4.2 Description de la technique de visibilité

Malgré l'introduction des techniques hiérarchiques et d'agrégations, la radiosité demeure toujours un processus très intensif en temps de calcul, de nombreuses heures pouvant être nécessaires pour obtenir une image de qualité d'une scène complexe. Lorsqu'on analyse ce processus, on s'aperçoit que le coût majeur est le calcul des facteurs de forme et plus précisément de la visibilité qui peut atteindre plus de 99% du coût total du rendu.

Il y a cependant beaucoup de cohérence entre ces requêtes, qui peut être exploitée afin d'accélérer le processus en diminuant le nombre d'échanges d'énergie lumineuse ainsi que le nombre de requêtes de visibilité. Par exemple, on peut éviter de nombreux échanges entre les éléments de deux pièces séparées si l'on peut détecter qu'il y a un mur entre ces éléments bloquant entièrement tous les échanges entre ces deux éléments. De même, on peut éviter de nombreuses requêtes de visibilité entre deux régions d'une scène, par exemple deux murs d'une même pièce, si l'on peut détecter qu'il n'y a aucun objet entre eux, c'est-à-dire qu'il sont mutuellement complètement visibles. On a pu observer dans nos différentes scènes test que le premier cas mentionné, c'est-à-dire les échanges complètement bloqués comptent pour 40 à 80% de tous les échanges et que le second cas compte pour 20 à 60% des échanges.

Dans cette section est présenté un nouvel algorithme de visibilité n'introduisant aucune erreur dans les échanges d'énergie lumineuse et permettant de détecter les deux

types d'événements mentionnés ci-haut pour diminuer les coûts de la détermination de la visibilité. Dans un premier temps, deux tests simples permettant de déceler les cas triviaux de visibilité complète et d'occlusion totale sont présentés. Puis ensuite, leur utilisation au coeur de l'algorithme de radiosité présenté dans la section 4.1.2 est décrite en détail.

4.2.1 Visibilité complète

L'utilisation de la visibilité complète⁵ a été exploitée dans quelques techniques de visibilité [HW91, MWCF90] afin de réduire leur coût total. Cet événement de visibilité est détecté entre deux groupes d'éléments (triangles, agrégats de faces ou de volumes) lorsque aucun autre élément de la scène s'interpose entre ces deux groupes. Ces groupes sont alors considérés comme étant mutuellement complètement visibles et aucune autre requête de visibilité n'est requise entre ces groupes. Il faut noter toutefois, qu'il peut y avoir du blocage à l'interne d'un groupe (i.e. du groupe sur lui-même). Pour détecter la visibilité complète, un tunnel est construit entre les deux groupes, puis celui-ci est testé contre la scène afin de déterminer s'il est vide.

Construction du tunnel

Le type de tunnel employé est le même que celui décrit par Haines et Wallace [HW91]. Alors pour plus de précisions techniques, les lecteurs devraient se rapporter à l'article original.

Dans un premier temps, les boîtes englobantes alignées sur les axes de chacun des agrégats⁶ sont récupérées. Ensuite un volume minimal délimitant la région désirée est formé. Pour ce faire, on utilise encore une fois, une boîte alignée sur les axes qui englobe de façon minimale les boîtes des deux agrégats. Finalement, la dernière étape consiste à réduire le volume pour définir l'espace interne du tunnel par un ensemble de plans. Un maximum de huit plans sont créés où chacun repose sur au moins une arête de chacune des deux boîtes des agrégats. Ainsi une des trois composantes de la normale de chaque plan est nulle. Chacune des étapes de construction du tunnel est démontrée à la figure 4.10.

⁵La visibilité complète désigne une visibilité sans aucune occlusion.

⁶Pour fin de simplicité, on dénote par agrégat soit un élément, un agrégat de faces ou de volumes selon le cas.

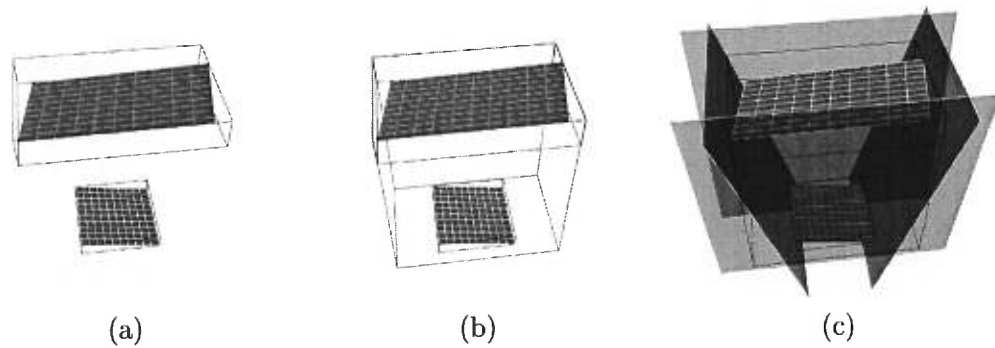


FIG. 4.10: Étapes de la création d'un tunnel. a) Agrégats de faces et leurs boîtes englobantes; b) Ajout de la boîte englobante du tunnel; c) Ajout de plans définissant le tunnel

Détection de la visibilité complète

Pour détecter rapidement l'état du tunnel, c'est-à-dire vérifier s'il est vide ou non, on a besoin d'une hiérarchie de volumes englobants de la scène. La hiérarchie de boîtes alignées sur les axes créée au chapitre précédent est traversée récursivement pour déceler la présence d'éléments à l'intérieur du tunnel. Chaque boîte en commençant par la racine est testée contre le tunnel pour déterminer le cheminement de la traversée de la hiérarchie. Trois cas peuvent survenir :

- La boîte est totalement à l'intérieur du tunnel. La traversée complète de la hiérarchie peut arrêter puisque des éléments sont présents dans le tunnel.
- La boîte est à l'extérieur du tunnel. La traversée de cette branche de la hiérarchie peut être arrêtée puisque aucun élément contenu dans celle-ci est dans le tunnel.
- La boîte intersecte partiellement le tunnel. Dans le cas où elle contient seulement un élément (un triangle) celui-ci est testé contre le tunnel. S'il intersecte ou qu'il est complètement à l'intérieur du tunnel, la traversée est finie; sinon la traversée de cette branche de la hiérarchie est terminée et l'on poursuit dans une autre. Dans le cas où la boîte contient une hiérarchie d'éléments, chacun de ses deux enfants est testé récursivement de la même façon.

Si toute la hiérarchie est traversée sans avoir conclu qu'au moins un élément intersecte le tunnel, alors le tunnel est vide et les deux groupes sont complètement visibles entre eux.

Pour définir l'état d'une boîte par rapport au tunnel, on la compare initialement à la boîte englobante du tunnel. Lorsque la boîte est extérieure, il s'agit d'un rejet trivial. Dans le cas contraire, la boîte doit être testée contre tous les plans formant le tunnel. Aussitôt que cette dernière est complètement extérieure à un plan, la boîte est externe au tunnel. Pour qu'elle soit considérée interne, elle doit être contenue à l'intérieur de chaque plan. Il n'est pas nécessaire de tester chaque sommet de la boîte contre un plan mais seulement celui le plus près du plan. Comme les boîtes sont alignées sur les axes, le sommet à tester peut être prédéterminé.

4.2.2 Occlusion totale

Déterminer tous les événements de visibilité résultant de la combinaison de plusieurs bloqueurs, ce qui est communément appelé la fusion de bloqueurs, est une tâche complexe. Il est nettement plus simple et plus rapide de déterminer si un seul bloqueur convexe bloque entièrement le tunnel formé précédemment entre les deux groupes d'éléments. On verra plus loin que malgré la simplicité du test et que seul l'occlusion d'un groupe connecté et convexe est prise en considération, un haut taux de blocage est identifié menant à des gains substantiels. Pour déterminer si un groupe d'éléments est complètement caché par rapport à un deuxième groupe, on utilise le même tunnel que pour le test de visibilité complète. Toutefois, on ne se sert pas des plans ni de la boîte englobante du tunnel mais des segments définissant son contour. Ces segments sont ceux rejoignant chacun des sommets respectifs de chacune des deux boîtes englobantes alignées sur les axes des groupes d'éléments. Ils sont au nombre de quatre à huit selon le positionnement mutuel des deux boîtes. Un exemple peut être observé à la figure 4.11.

Détection de l'occlusion totale

La détermination d'un blocage du tunnel provoqué par un bloqueur convexe est obtenu en testant tous les segments de contour du tunnel contre le bloqueur. Si tous les segments intersectent le même bloqueur, les deux groupes d'éléments sont garantis de ne pas voir la moindre portion du groupe opposé. Toutefois, si un seul des segments n'intersecte pas le bloqueur, même si un autre bloqueur pourrait bloquer la portion restante (fusion de bloqueurs), on ne peut rien conclure à l'exception que les deux régions sont possiblement bloquées. Voici donc les étapes de l'algorithme servant à

déterminer l'occlusion d'un tunnel :

1. Un tunnel (section 4.2.1) est construit à partir des boîtes englobantes (alignées sur les axes de la scène) des deux éléments.
2. Quatre à huit segments (dépendamment des positions respectives des éléments) sur le contour du tunnel sont calculés.
3. Le tunnel est testé contre la hiérarchie des bloqueurs extraits
 - (a) On lance un segment⁷ dans la liste de bloqueurs.
 - (b) Pour chaque bloqueur intersecté, tous les autres segments du contour du tunnel sont testés.
 - Aussitôt qu'un segment n'intersecte pas le bloqueur, on recommence à intersecter le premier segment avec un autre bloqueur de la liste.
 - Si tous les rayons intersectent le bloqueur, on peut garantir l'occlusion totale.

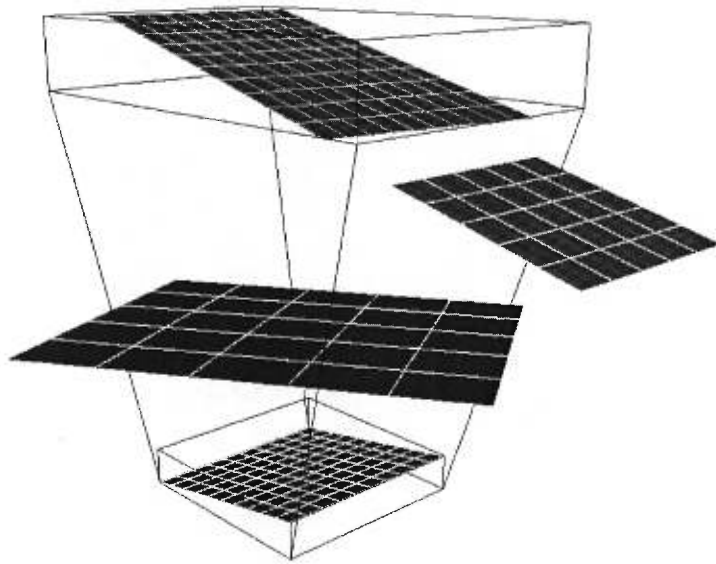


FIG. 4.11: Le tunnel formé entre deux groupes d'éléments est testé pour occlusion totale. L'agrégat du haut bloque partiellement le tunnel, tandis que celui du bas le bloque totalement puisqu'il intersecte tous les segments.

Puisque plusieurs segments doivent être testés, l'algorithme utilise une approche itérative [Smi98] pour traverser efficacement la hiérarchie de boîtes englobantes au lieu

⁷En fait, on effectue l'équivalent de lancer un rayon d'ombre correspondant au segment.

d'une approche récursive plus traditionnelle. Des gains de l'ordre d'un facteur de trois à quatre ont pu être observés dans cette portion de l'algorithme.

4.2.3 Système de visibilité

Les deux tests décrits précédemment pour déceler rapidement à coût modique les événements de visibilité complète et d'occlusion totale peuvent être utilisés avec profit dans tous les types d'algorithmes de radiosité mais leur efficacité est d'autant plus importante s'ils sont utilisés dans un algorithme hiérarchique tel que décrit à la section 4.1.2. La nature hiérarchique de l'algorithme permet aux tests de visibilité d'obtenir une réponse positive à un haut niveau de la hiérarchie et donc augmente leur efficacité.

Les deux tests sont incorporés au niveau de l'oracle (voir l'algorithme 10) permettant à celui-ci d'arrêter l'échange d'énergie en cas d'occlusion totale et d'effectuer l'échange sans test de visibilité en cas de visibilité complète. Ces tests permettent aussi de propager l'information au cours des subdivisions subséquentes afin d'éviter de nouvelles requêtes. Alors voici les différentes étapes introduites dans l'oracle :

- En présence d'au moins un agrégat de volume (l'émetteur et/ou le receveur), on le subdivise et aucun échange d'énergie n'a lieu. La méthode de Sillion *et al.* [SDS95] d'échanges entre volumes pourrait très bien être utilisée, toutefois la qualité du rendu pourrait être inférieure. Dans certains cas cela en vaudrait la peine pour les gains en vitesse lorsque l'agrégat contient un nombre élevé de petits éléments distribués plus ou moins uniformément comme les feuilles d'un arbre, ou encore que l'agrégat est très loin de l'autre groupe.
- En présence d'un agrégat de faces jugé insuffisamment planaire (i.e. la variation de normales est considérée trop élevée), on subdivise encore une fois. Dans le cas contraire, on effectue un test d'élimination de faces arrière. Si l'émetteur et le receveur sont opposés alors la récursion arrête car il ne peut y avoir d'échanges.
- Par la suite, la visibilité complète et l'occlusion totale entre deux éléments sont testées, mais seulement à chaque quatre niveaux de subdivision dans la hiérarchie. Ceci permet au tunnel de rétrécir et de changer de contexte (changement de position et de contenu) suffisamment pour augmenter la probabilité d'obtenir une réponse positive (complètement visible ou complètement bloqué) et par le

fait même, de réduire le coût total du calcul.

- L’occlusion totale est moins coûteuse à calculer et donc effectuée en premier. Si l’occlusion totale est détectée le processus de subdivision est arrêté et aucun échange d’énergie n’a lieu.
- Si l’occlusion totale n’est pas détectée alors on passe au test de visibilité complète. On pourrait penser que si une occlusion partielle a été détectée à l’étape précédente, on a nul besoin de tester la visibilité complète. Cependant on a observé qu’en marquant les boîtes englobantes à l’extérieur du tunnel (expliqué plus bas) on obtient de meilleurs gains même si on sait a priori que le résultat du test sera négatif. Lorsque le test est positif, l’information est transmise par l’oracle lors des subdivisions subséquentes afin d’éviter de nouvelles requêtes de visibilité.
- Lorsque l’oracle décide que le niveau de subdivision atteint engendrera une erreur suffisamment petite pour effectuer l’échange d’énergie lumineuse et que la visibilité complète n’a pas été détectée, alors on se rabat sur une technique conventionnelle pour évaluer la visibilité entre l’émetteur et le receveur. Dans notre cas on génère n rayons aléatoirement entre les deux groupes [HSA91].

Algorithme 10: Oracle2(p , q , ref(v))

```

Entrée :  $p$  : élément émetteur
          $q$  : élément receveur
Données : seuil : erreur minimum que l'on veut permettre
         tailleMinimum : taille minimum que peut avoir un élément
          $v \in \{VisibilitéComplète, OcclusionTotale, Incertain\}$ 

si  $v \neq VisibilitéComplète$  alors
  |  $v \leftarrow CalculVisibilité$ ;
  | si  $v = OcclusionTotale$  alors retourner aucun;
fin
erreur  $\leftarrow CalculErreur()$ ;
si erreur > seuil alors
  | si Taille( $p$ ) > tailleMinimum alors
  | | Subdivise( $p$ );
  | | retourner  $p$ ;
  | sinon
  | | Subdivise( $q$ );
  | | retourner  $q$ ;
  | fin
sinon
  | retourner aucun;
fin

```

Amélioration du test de visibilité complète

Pour optimiser les tests de visibilité complète et de lancer de rayons, tous deux utilisant la même hiérarchie de la scène, on peut marquer toutes les boîtes étant détectées comme extérieures au tunnel pour qu’elles ne soient pas testées à nouveau lors de subdivisions subséquentes de ce tunnel. Une liste des boîtes marquées est gardée afin de pouvoir les “insérer” à nouveau dans la hiérarchie lorsque l’algorithme remonte au niveau des parents du noeud où le marquage se produit.

4.3 Résultats et analyse

4.3.1 Statistiques sur nos scènes test

Les résultats pour notre méthode de visibilité présentée à la section précédente sont résumés dans le tableau 4.1. L’ensemble des scènes sélectionnées démontre une grande variété de contextes de visibilité. Une image de chacune de ces scènes apparaît à la figure 4.13 à la page 64, ainsi que le nombre de triangles la composant et le nombre de bloqueurs planaires extraits. Certaines de ces images ont reçu une correction gamma de leurs couleurs pour mieux exposer leur contenu.

Les scènes A et E présentent un arrangement de pièces ayant très peu d’échanges d’énergie entre elles puisque étant très fermées. Au contraire dans la scène B, toutes les pièces peuvent voir une bonne portion des autres, facilitant l’échange d’énergie (la scène B est constituée d’une matrice de 10×10 pièces). La scène C contient un haut taux d’occlusion mais est constituée d’une seule pièce ouverte. Cette dernière configuration est intéressante puisqu’elle serait difficilement optimisable par la technique des portails⁸ [TS91]. La scène D est un exemple de pièce avec très peu d’occlusion. En fait seule la table et les contours des lumières provoquent un peu d’occlusion. Finalement, la scène F est un mélange d’occlusion totale et de visibilité complète ; au total, ces deux types d’événements comptent pour 99% des événements de visibilité dans cette scène.

Afin de tester l’efficacité du nouvel algorithme de visibilité, trois techniques sont testées sur chacune des scènes. La première, appelée *normale*, est l’algorithme de base décrit à la section 4.1.2 sans aucun ajout et servant de point de référence pour les tests

⁸Puisque la scène n’est qu’une aire ouverte, elle ne possède aucun “trou” pouvant devenir un bon portail.

subséquents. La seconde technique, *visibilité*, est la même technique que la *normale* mais avec l'addition du test de visibilité complète. La technique *occlusion* quant à elle, est celle décrite en détail dans la section 4.2, et elle est donc le nouvel algorithme dans sa version intégrale. Les données sont aussi séparées selon l'itération effectuée. Une itération de radiosité consiste à propager la radiosité de la scène à la scène en descendant avec l'oracle la hiérarchie complète. Une première itération correspond à l'illumination directe des sources, deux à la première réflexion de l'énergie et ainsi de suite.

Le temps total affiché est en secondes, et provient d'un PC Athlon de 256 MB de mémoire vive et roulant linux à 600 Mhz.

Pour chaque scène, le tableau 4.1 donne le nombre d'échanges d'énergie (en millions) entre les éléments (agrégats de faces ou triangles) et la quantité d'entre eux qui ont été complètement bloqués, c'est-à-dire que tous les rayons lancés pour évaluer la visibilité ont été intersectés par une surface. La quantité totale de rayons lancés et la quantité de ceux-ci qui furent bloqués par occlusion totale ou partielle sont aussi affichées. Le nombre de tunnels construits est indiqué et représente par la même occasion, le nombre de requêtes de visibilité complète et d'occlusion totale effectuées.

La colonne *occlusion totale* est divisée en deux parties : premièrement, le pourcentage des échanges qui furent complètement bloqués et ensuite, le pourcentage d'échanges complètement bloqués en comparaison avec le maximum possible (le premier terme). Le premier terme est en fait un ensemble englobant (borne supérieure sur la quantité d'éléments) de tous les échanges bloqués de la scène et ceci parce que cette statistique a été recueillie par échantillonnage en lancer de rayons. Donc certains échanges considérés comme étant complètement bloqués car tous les rayons furent intersectés peuvent ne pas être réellement entièrement bloqués. La colonne *visibilité complète* représente quant à elle, le pourcentage d'échanges qui furent effectués sans requérir au lancer de rayons car un tunnel parent à l'échange fut détecté comme vide et donc complètement visible.

4.3.2 Analyse

Comme on peut le remarquer à partir des temps de calcul obtenus, l'utilisation du nouvel algorithme de visibilité résulte en une accélération d'un facteur variant entre 1.3 à 11 pour toutes nos scènes test, et avec des gains en moyenne de plus de 2.5. Dans

aucun cas, notre technique n'a démontré un impact négatif sur les performances. On peut toutefois noter que l'ajout du test d'occlusion totale a dans un cas (scène D) réduit légèrement (1%) les gains déjà apportés par le test de visibilité complète. Il faut noter que cette scène ne contient presque pas d'événements d'occlusion totale. Cependant on peut remarquer que le coût supplémentaire induit par ce test est très bas.

De plus on peut noter que les gains sont légèrement réduits pour l'éclairage indirect (itérations 2 et 3). Ces échanges d'énergie possédant beaucoup moins d'énergie se produisent à un plus haut niveau dans la hiérarchie de la scène. Une conséquence directe est que la taille du tunnel entre l'émetteur et le receveur augmente, et ainsi la taille relative entre le tunnel et les bloqueurs diminue, rendant ainsi moins fréquentes les occlusions totales. Par ailleurs, plus le tunnel est gros, plus il contiendra des éléments, ce qui réduit les cas de visibilité complète.

Bien que notre algorithme d'extraction de bloqueurs et notre test de visibilité ne permettent pas de fusion générale de bloqueurs, jusqu'à 93% (scène F) de toute l'occlusion possible fut considérée par nos bloqueurs. Un autre aspect intéressant mentionné précédemment est que ces tests sont tellement peu coûteux que même dans les cas où ils sont inutiles par faute de manque d'occlusion totale ou de visibilité complète dans la scène, leur impact sur le temps de calcul total est négligeable.

La proportion maximale d'occlusion totale calculée dans certaines scènes test peut paraître inférieure à ce qu'on peut s'attendre. Par exemple, la scène F a un facteur d'occlusion de 38% malgré que chacune des sources lumineuses voit moins que la moitié de la géométrie totale. Ce phénomène s'explique par le fait que la plupart des échanges d'énergie s'effectuent dans un périmètre très proche de l'émetteur. La contribution du facteur de forme dans la métrique d'erreur du raffinement *BFA* au niveau de l'oracle en est la cause. En fait, la quantité d'échanges est inversement proportionnelle au carré de la distance entre l'émetteur et le receveur. On observe effectivement à la figure 4.12 que la densité du maillage est beaucoup plus élevée près de l'émetteur.

Scène	Nombre d'itération	Technique d'occlusion	Temps total (sec)	Échanges d'énergie (M)	Échanges bloqués (M)	Rayons lancés (M)	Rayons bloqués (M)	Tunnels construits (M)	Occlusion totale (%)	Visibilité complète (%)
A	1	normale	428	4.7	1.78	37.9	15.8	0	38	46
		visibilité	271	4.7	1.78	20.3	15.8	3.0	73	
		occlusion	220	3.4	0.47	9.9	1.9	1.9		
B	1	normale	382	4.8	2.91	39.0	24.5	0	61	29
		visibilité	275	4.8	2.91	27.7	24.5	9.5	69	
		occlusion	173	2.8	0.86	11.3	8.1	3.4		
C	1	normale	199	1.20	0.64	12.07	6.96	0	53	19
		visibilité	141	1.20	0.64	9.80	6.96	0.98	81	
		occlusion	88	0.68	0.13	4.65	1.84	0.44		
	2	normale	430	2.40	1.95	23.72	20.84	0	81	2
		visibilité	435	2.40	1.95	23.33	20.84	6.96	75	
		occlusion	200	0.93	0.49	8.68	6.45	2.57		
3	normale	532	2.99	2.66	29.90	28.12	0	88	1	
	visibilité	553	2.99	2.66	29.66	28.07	8.99	77		
	occlusion	228	0.93	0.62	9.14	7.58	3.17			
D	1	normale	93	0.32	0.05	6.60	1.29	0	16	56
		visibilité	28	0.32	0.05	2.88	1.29	0.08	40	
		occlusion	25	0.30	0.02	2.31	0.73	0.07		
	2	normale	55	0.23	0.02	4.62	0.85	0	9	34
		visibilité	33	0.23	0.02	3.04	0.85	0.18	50	
		occlusion	34	0.22	0.02	2.95	0.80	0.17		
3	normale	44	0.17	0.02	3.38	0.98	0	12	24	
	visibilité	33	0.17	0.02	2.57	0.98	0.22	50		
	occlusion	34	0.16	0.02	2.48	0.93	0.22			
E	1	normale	273	2.29	1.41	18.37	12.02	0	62	18
		visibilité	158	2.29	1.41	15.14	12.02	1.02	81	
		occlusion	95	1.15	0.27	6.01	2.89	0.47		
	2	normale	118	0.98	0.40	7.84	3.74	0	41	26
		visibilité	76	0.98	0.40	5.81	3.74	0.74	58	
		occlusion	69	0.75	0.17	3.95	1.92	0.49		
3	normale	89	0.84	0.48	6.70	4.24	0	57	17	
	visibilité	73	0.84	0.48	5.53	4.23	0.96	65		
	occlusion	59	0.53	0.18	3.08	1.82	0.52			
F	1	normale	1017	11.13	4.26	89.04	34.30	0	38	61
		visibilité	413	11.13	4.26	34.92	34.30	2.28	93	
		occlusion	92	7.15	0.29	3.12	2.50	0.36		
	2	normale	639	5.63	2.41	45.06	19.66	0	43	53
		visibilité	256	5.63	2.41	21.07	19.66	1.50	66	
		occlusion	139	4.05	0.82	8.36	6.96	1.05		
3	normale	298	2.57	1.12	20.53	9.24	0	44	51	
	visibilité	126	2.57	1.12	10.11	9.24	0.78	53		
	occlusion	86	1.98	0.54	5.40	4.53	0.66			

TAB. 4.1: Visibilité complète et occlusion totale sur des scènes variées de la figure 4.13

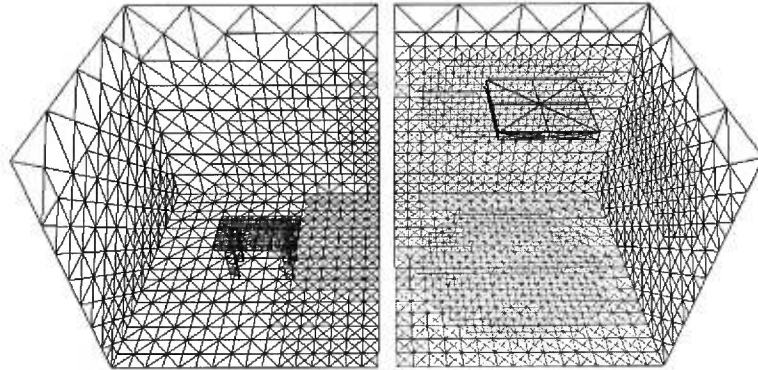


FIG. 4.12: La densité du maillage est plus élevée aux alentours de l'émetteur

4.4 Extension

4.4.1 Bloqueurs directionnels

Malgré les bons résultats obtenus et les nombreuses scènes pouvant en bénéficier, l'utilisation de bloqueurs planaires ne résout pas tous les problèmes de visibilité. Il est certain que les scènes architecturales typiques sont d'excellentes candidates pour les bloqueurs planaires avec leurs murs, plafonds, planchers, tables et autres surfaces de même acabit. Cependant, comme mentionné au chapitre précédent, de nombreuses surfaces non planaires peuvent elles aussi provoquer une quantité d'occlusion non négligeable et nous aimerions pouvoir exploiter ce potentiel.

Cette section décrit comment étendre l'algorithme de visibilité présenté dans ce chapitre pour tenir compte de l'occlusion provenant de surfaces non planaires avec l'aide de nos bloqueurs directionnels.

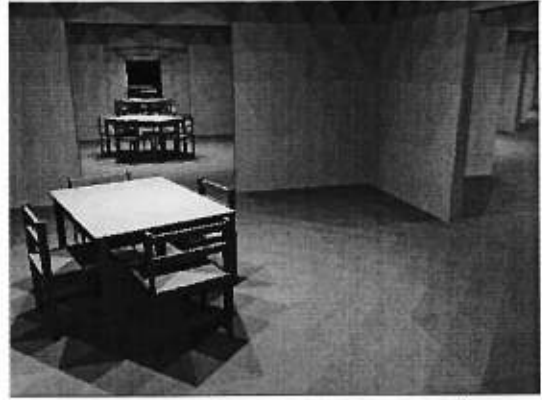
Modification de l'algorithme de visibilité

L'introduction de bloqueurs directionnels au coeur de l'algorithme de visibilité ne nécessite que des changements mineurs. En effet, seul le test d'occlusion totale doit être modifié pour tenir compte de cet ajout. Vue sa nature de bloqueur convexe et de surcroît, rectangulaire, le test d'intersection du bloqueur demeure inchangé. Cependant on doit tenir compte du champ d'action (pyramide de visibilité) du bloqueur.

Chaque bloqueur possède un drapeau permettant d'identifier rapidement ceux ayant



A : 37424 triangles (204 bloqueurs)



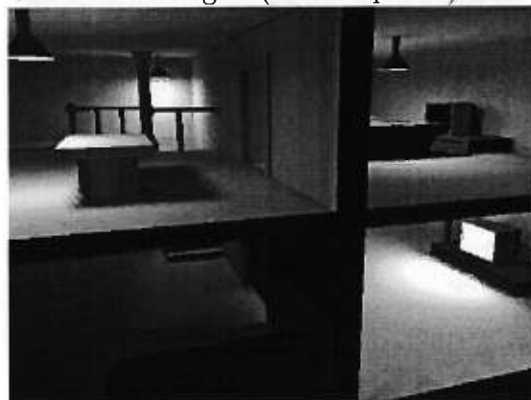
B : 83600 triangles (1140 bloqueurs)



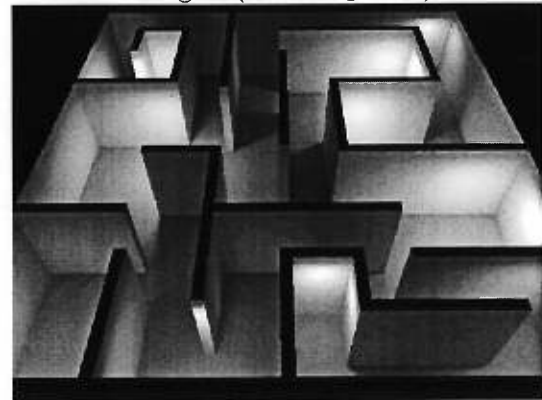
C : 51296 triangles (549 bloqueurs)



D : 741 triangles (119 bloqueurs)



E : 14453 triangles (549 bloqueurs)



F : 642 triangles (203 bloqueurs)

FIG. 4.13: Scènes test

une pyramide de visibilité couvrant tout l'hémisphère, donc nos bloqueurs planaires. Ces derniers ne nécessitent aucun test de validation. Par contre, avant d'utiliser un bloqueur directionnel, on doit vérifier sa validité par rapport au test de visibilité. Dans le cas de l'occlusion totale d'un tunnel contre un bloqueur directionnel, on doit s'assurer que l'un des deux agrégats est contenu entièrement dans la pyramide de visibilité du bloqueur. Si c'est le cas, l'occlusion provoquée par le bloqueur sur le tunnel est conservatrice et donc valide, mais dans le cas contraire l'intersection n'est pas vérifiée, et ce bloqueur directionnel ne peut pas être utilisé.

Pour optimiser le test d'inclusion des agrégats dans la pyramide de visibilité, il n'est pas nécessaire de tester les deux agrégats, mais seules les extrémités⁹ des segments du tunnel suffisent. Pour être considérées valides, toutes les extrémités des segments d'un même côté doivent être à l'intérieur des cinq plans définissant la pyramide.

On peut aussi noter que deux bloqueurs directionnels sont formés pour chaque surfaces. Ainsi au lieu d'utiliser simplement le dessous de la boîte englobante orientée de la surface pour construire le bloqueur directionnel, on utilise aussi le dessus pour en construire un second d'orientation opposée. Ceci permet de mieux couvrir l'espace d'occlusion.

Résultats

Les résultats pour notre méthode utilisant les bloqueurs directionnels sont présentés dans le tableau 4.2. Les différentes scènes utilisées sont démontrées à la figure 4.15 à la page 68. Tous les calculs ont été effectués seulement pour l'illumination directe. En plus de tester les scènes avec les trois techniques précédentes (*normale*, *visibilité*, *occlusion*), une variante est effectuée lors de la technique d'occlusion complète. Deux facteurs d'amplitude α (voir équation 3.14 à la page 31) ont été testés. Plus le facteur d'amplitude est grand, plus le bloqueur directionnel est grand, mais sa pyramide de visibilité en est diminuée. L'effet de ce facteur est démontré à la figure 4.14 où l'on peut remarquer qu'avec un α de 3 une plus grande zone¹⁰ est bloquée puisque le bloqueur directionnel est plus grand.

Encore une fois, les résultats proviennent d'un PC Athlon de 256 MB de mémoire

⁹Les extrémités sont les deux sommets de chaque segment du tunnel utilisés lors du test d'occlusion totale.

¹⁰La taille de la zone occludée peut être discernée par la subdivision moins importante sur le plancher.

vive et roulant linux à 600 Mhz. Les résultats obtenus sont similaires à ceux provenant des premières scènes (A à F) n'utilisant pas de bloqueurs directionnels. Ce qui est intéressant cependant, c'est qu'une bonne proportion de l'occlusion provient des bloqueurs directionnels. Un gain d'un facteur variant entre 3.7 à 4.3 pour le temps de calcul est ainsi obtenu. On peut noter qu'un facteur d'amplitude α de 3 procure de meilleurs résultats dans les scènes testées.

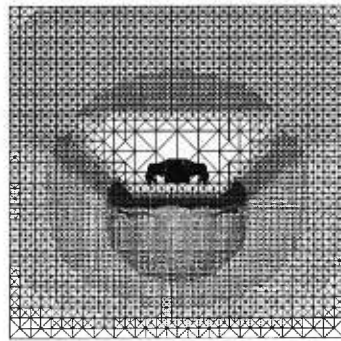
Scène	Technique d'occlusion	Temps total (sec)	Échanges d'énergie (M)	Échanges bloqués (M)	Rayons lancés (M)	Rayons bloqués (M)	Tunnels construits (M)	Occlusion totale (%)
G	normale	23.0	0.29	0.10	2.29	0.78	0	33 81
	visibilité	14.8	0.29	0.10	1.29	0.78	0.084	
	occlusion, $\alpha = 1$	9.1	0.23	0.03	0.80	0.29	0.067	
	occlusion, $\alpha = 3$	6.2	0.21	0.02	0.67	0.16	0.052	
H	normale	22.0	0.24	0.08	1.92	0.64	0	31 70
	visibilité	8.1	0.24	0.08	0.81	0.64	0.060	
	occlusion, $\alpha = 1$	5.9	0.19	0.03	0.42	0.26	0.041	
	occlusion, $\alpha = 3$	5.6	0.19	0.02	0.38	0.22	0.039	
I	normale	430.0	3.28	1.23	26.21	10.52	0	38 61
	visibilité	148.4	3.28	1.23	12.32	10.52	0.88	
	occlusion, $\alpha = 1$	108.8	2.64	0.60	7.25	5.45	0.60	
	occlusion, $\alpha = 3$	100.2	2.53	0.48	6.32	4.52	0.54	

TAB. 4.2: Visibilité complète et occlusion totale sur des scènes variées de la figure 4.13

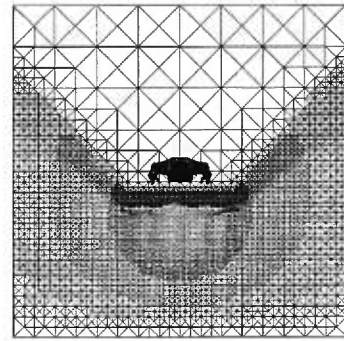
4.4.2 Bloqueurs volumiques

L'utilisation de bloqueurs volumiques (ceux construits par la technique d'Andujar *et al.* [ASVN00]) à l'intérieur de la technique de visibilité décrite précédemment se fait sans aucun problème. Puisque ces bloqueurs sont convexes, il suffit d'utiliser un test d'intersection adapté à la forme du bloqueur lorsque celui-ci est vérifié contre un segment du tunnel.

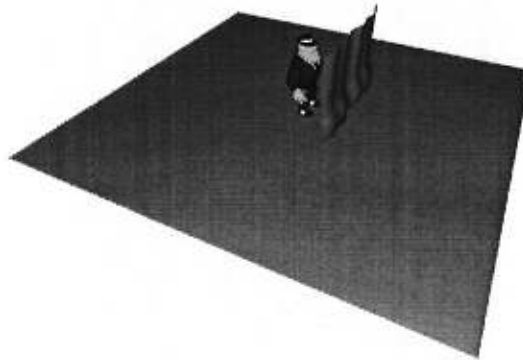
Pour ce qui est des bloqueurs volumiques non convexes, ils ne peuvent être utilisés directement par l'algorithme de visibilité. Toutefois, on pourrait les subdiviser en sous-volumes convexes qui eux pourraient être utilisés sans problème. Pour augmenter l'efficacité de ces nouveaux bloqueurs, on pourrait les "gonfler" après leur division jusqu'à ce qu'ils atteignent les limites de l'objet initial.



(a) subdivisions avec $\alpha = 1$



(b) subdivisions avec $\alpha = 3$

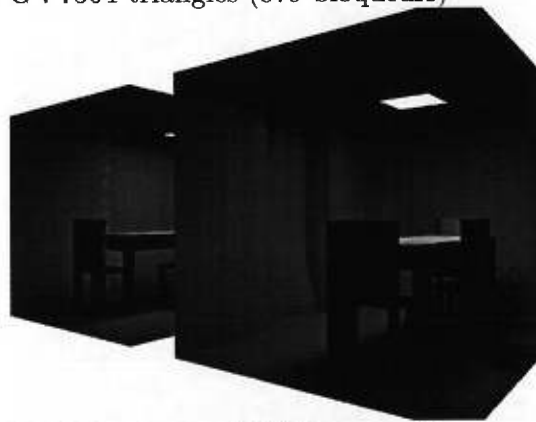


(c) scène initiale

FIG. 4.14: Résultat de différents facteurs d'amplitude α sur une scène



G : 7564 triangles (879 bloqueurs)



H : 808 triangles (179 bloqueurs)



I : 11754 triangles (1005 bloqueurs)

FIG. 4.15: Scènes test

Chapitre 5

Lumières étendues

5.1 Description du problème

Ce chapitre se veut un premier pas afin d'étendre l'utilisation de nos bloqueurs pour accélérer les requêtes de visibilité. Par conséquent le nouvel algorithme n'est que brièvement abordé et plusieurs améliorations sont suggérées afin d'augmenter ses performances.

On présente ici une technique d'accélération pour le calcul d'ombrage (visibilité) de sources étendues lors du rendu par tracé de rayons. Au chapitre précédent, les bloqueurs ont permis d'accélérer la vitesse du calcul de la radiosité hiérarchique pour des scènes de complexités variables. Le contexte présenté dans ce chapitre, bien que semblable au précédent à bien des égards (le calcul de l'éclairage direct est aussi considéré en radiosité), demeure néanmoins légèrement différent. En effet, seul l'éclairage de la partie visible de la scène à partir de la caméra (i.e. visible dans l'image) doit être calculée. De plus, seulement l'éclairage direct à partir de sources étendues est abordé, donc aucun éclairage indirect n'est calculé.

Il est à noter que bien que la technique est présentée dans le cadre d'un algorithme de tracé de rayons, elle peut être aussi bien utilisée pour le balayage de lignes et d'autres techniques de rendu similaires.

5.2 Description de la méthode

5.2.1 Technique standard

Plusieurs méthodes [ARHM00, HDG99, CW93a] permettent de simuler l'éclairage obtenu directement d'une source lumineuse étendue. Cependant, le lancer de rayons distribué [CPC84] demeure tout de même la façon standard de calculer ces ombres floues. Elle permet de calculer l'intensité lumineuse d'un point d'une surface en provenance d'une lumière étendue. Pour ce faire, on génère aléatoirement un certain nombre de points distribués uniformément sur la source lumineuse, et un rayon d'ombre est lancé pour chacun de ces points afin de déterminer la visibilité mutuelle entre la source et le point de la surface. S'ils sont visibles, la contribution de l'illumination au point de la surface est calculée par rapport à son énergie et son facteur de forme (orientation et distance). Plus de rayons (points) sont utilisés, plus la solution converge.

5.2.2 Technique avec bloqueur

La technique proposée s'effectue seulement sur la base d'un point d'une surface éclairée par une lumière étendue (similairement au lancer de rayons distribué [CPC84]). Pour chaque point d'une surface dont on veut calculer l'intensité lumineuse et pour chaque lumière étendue, on crée un tunnel contenant la lumière et le point à éclairer. Ensuite, le tunnel est testé pour la visibilité complète et l'occlusion totale tel qu'expliqué au chapitre précédent. Si le tunnel est vide, alors on calcule l'illumination de la façon standard mais sans lancer de rayons d'ombre puisque le point "voit" toute la lumière. Par contre, si le tunnel est totalement bloqué, la lumière ne contribue pas à l'illumination du point et on peut passer à la lumière étendue suivante s'il y a lieu. Si le tunnel est partiellement bloqué, on utilise tout simplement la technique standard avec le lancer de rayons d'ombre pour calculer la visibilité.

5.3 Résultats

Le tableau 5.1 résume les résultats obtenus par notre méthode sur un PC roulant linux à 600 MHz. Les scènes ainsi que les résultats visuels obtenus par notre algorithme sont illustrés à la figure 5.1. Encore une fois, nos scènes test varient de façon importante en complexité. Ainsi la scène B contient 741 triangles alors que la scène D en contient

près de 200 000 et plus de 700 lumières étendues. Deux techniques sont testées pour chaque scène : la technique *normale* est celle décrite à la section 5.2.1, soit le lancer de rayons distribué; la technique *occlusion* ajoute les détections rapides de visibilité complète et d'occlusion totale, telles que décrites dans ce mémoire.

Lorsque pour chaque point on échantillonne chaque source lumineuse à l'aide de 10 rayons, on dénote une accélération du temps total de rendu variant de 1.67 à 2.45. Avec 100 rayons par point, l'accélération observée varie de 2.56 à 4.94. Il est à noter que la dernière scène n'a pas été testée avec un échantillonnage de 100 rayons par pixel à cause de contraintes temporelles. Le temps estimé pour la technique *normale* est d'environ 260000 secondes (72 heures) et d'un peu plus de 15 heures pour la technique *occlusion*. On peut observer à la figure 5.1 que l'ombre calculée avec 10 rayons par pixel est assez bruitée tandis qu'avec 100 rayons par pixel le bruit est presque imperceptible. Il est donc important de lancer un grand nombre de rayons pour obtenir une image de qualité.

Scène	Technique de rendu	Temps de rendu 10 rayons/pixel (sec)	Temps de rendu 100 rayons/pixel (sec)
A	normale	29.42	274.87
	occlusion	12.02	55.54
B	normale	88.24	840.88
	occlusion	50.65	270.49
C	normale	148.58	1452.64
	occlusion	88.59	564.75
D	normale	25943.4	
	occlusion	11678.0	

TAB. 5.1: Visibilité complète et occlusion totale sur les différentes scènes de la figure 5.1

5.4 Améliorations

La technique décrite dans ce chapitre n'est qu'une ébauche de ce que devrait être l'algorithme complet. Plusieurs améliorations peuvent être apportées afin d'augmenter encore plus son efficacité. Dans cette section, quatre de ces améliorations sont présentées, basées sur l'agrégation et le raffinement adaptatif.

5.4.1 Regroupement en agrégats de lumières

Les lumières ne devraient pas être testées une par une pour chaque point à illuminer, mais plutôt hiérarchiquement lors des tests de visibilité complète et d'occlusion totale.

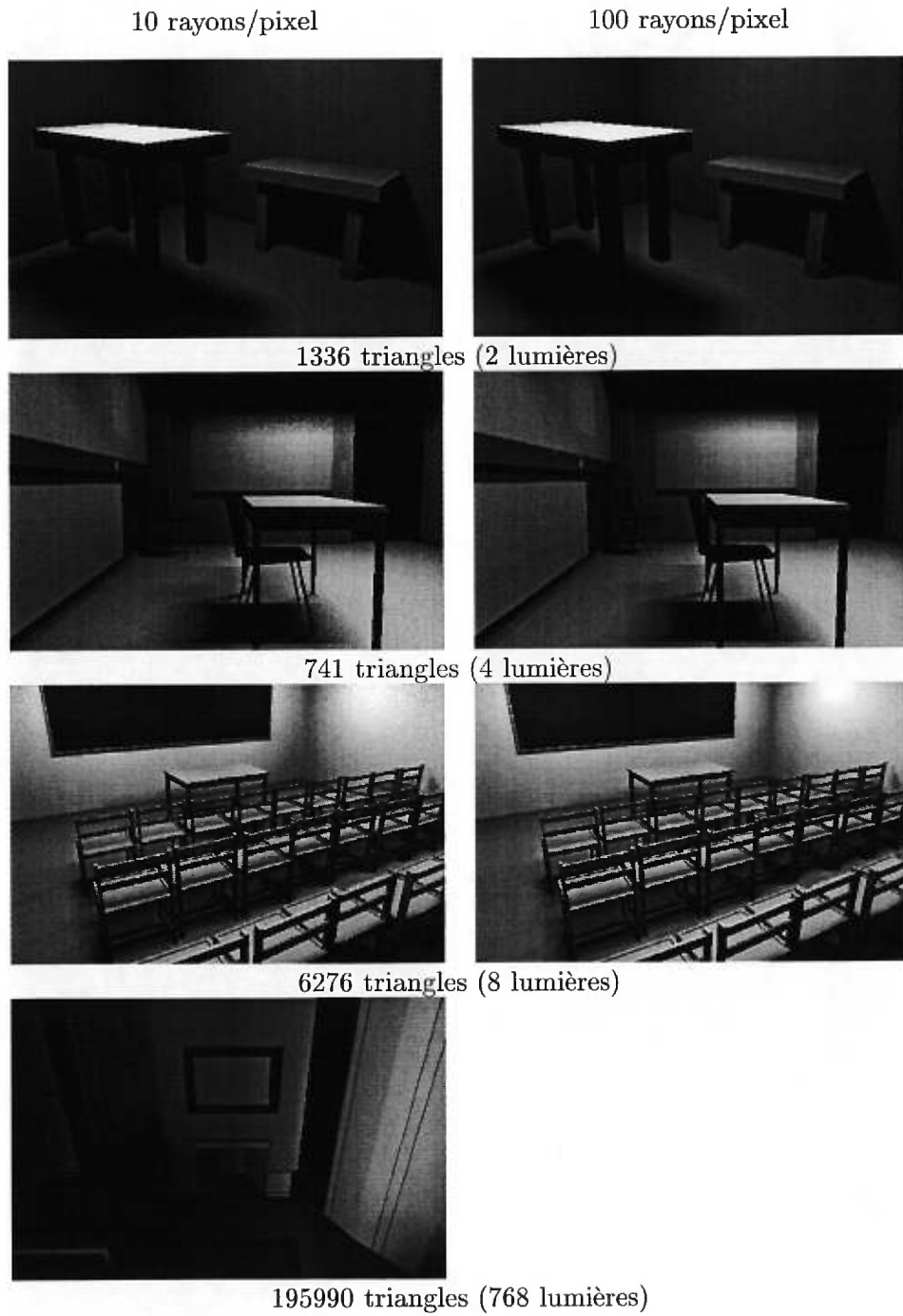


FIG. 5.1: Scènes test

Ainsi, les lumières seraient regroupées en agrégats et l'algorithme pourrait être capable d'éliminer en un test une grande quantité de lumières n'éclairant pas le point, ou encore de trouver à haut niveau qu'un groupe de lumières voit complètement le point.

5.4.2 Subdivision de lumières

Une grande source lumineuse étendue est dans bien des cas seulement partiellement visible, ce qui peut résulter en une perte de vitesse dans le temps de calcul puisque les deux tests de visibilité échoueraient dans la majorité des configurations. Dans un tel cas, il faudrait diviser la source et tester chacune de ses parties récursivement.

5.4.3 Regroupement par surfaces ou par régions d'image

Pour exploiter encore plus la cohérence afin de maximiser les gains, il serait préférable de ne pas tester tous les points des surfaces séparément. Les tests de visibilité envers les lumières pourraient être faits par surface. Pour ne traiter que les surfaces visibles on pourrait faire une passe de rendu mais sans le calcul d'éclairage, puis effectuer les tests de visibilité seulement sur les surfaces visibles. Ou encore, au fur et à mesure du rendu de la scène, les nouvelles surfaces sont testées et pour chaque nouveau point, on vérifie si sa surface a déjà été testée.

Similairement, au lieu de regrouper les points par surface, on pourrait regrouper les points par région voisine de l'image rendue, profitant ainsi de la cohérence image.

Toutes ces techniques devraient améliorer encore plus les gains observés pour le rendu de l'illumination de sources étendues. La généralité de nos bloqueurs et leur exploitation efficace ouvre plusieurs avenues d'applications.

Chapitre 6

Conclusion

Au cours de ce mémoire nous avons présenté des algorithmes efficaces pour créer des bloqueurs planaires, directionnels et volumiques. Les bloqueurs planaires représentent bien l'occlusion des scènes de type architectural où de grands bloqueurs peuvent être extraits pour les murs, les planchers, les plafonds et encore bien d'autres surfaces. En fait, dans ces scènes, on détecte à l'aide de nos bloqueurs planaires entre 40% à 80% d'occlusion totale en radiosit . L'introduction de bloqueurs directionnels permet de g n raliser le concept de bloqueurs planaires pour des surfaces quelconques mais continues. La construction de ces bloqueurs utilise l'agr gation de faces et ainsi permet d'extraire des bloqueurs dans une sc ne   partir seulement d'une soupe de triangles, ce qui rend les algorithmes pr sent s facilement utilisables dans presque toutes les sc nes.

Ensuite, l'efficacit  de tels bloqueurs pour acc l rer les requ tes de visibilit  fut d montr e dans deux contextes, soit la radiosit  et le calcul d' clairage direct par sources  tendues. On a pr sent  un algorithme conservateur d' limination de surfaces cach es implant  dans un algorithme de radiosit  hi rarchique par agr gation de faces. Cet algorithme ainsi que la construction de bloqueurs planaires furent l'objet d'une publication [LP00]. Dans cet algorithme nous avons exploit  efficacement les occurrences de visibilit  compl te et d'occlusion totale procurant ainsi un arr t t t lors du calcul de visibilit  et ce, sans introduire d'erreurs. Ceci entra ne des  changes d' nergie de haute qualit    un plus bas co t de calcul.

L'acc l ration se fait   l'aide d'une hi rarchie de bloqueurs que l'on intersecte efficacement avec les segments formant le tunnel entre deux r gions. La taille de cette hi rarchie peut  tre sp cifi e par un utilisateur permettant ainsi   l'algorithme de bien

fonctionner même avec une mémoire limitée. Cette méthode de visibilité fut testée et analysée sur plusieurs scènes de complexités variables. Dans tous les cas, notre structure a prouvé son efficacité (accélération jusqu'à un facteur 11), même dans les scènes où la visibilité complète et l'occlusion totale n'étaient pas apparentes. La méthode proposée est donc simple à implantée, demande peu de mémoire, procure un bon facteur d'accélération, et n'introduit pas d'erreurs dans la solution.

Dans un deuxième temps, l'utilisation de bloqueurs fut introduite dans un algorithme de lancer de rayons afin d'accélérer le calcul de l'illumination directe de sources lumineuses étendues. Encore une fois, malgré la simplicité de la méthode, des gains intéressants furent apportés (jusqu'à un facteur 5). Cette méthode ne présente qu'un premier pas et ainsi plusieurs améliorations prometteuses sont suggérées.

Malgré que les techniques présentées au cours de ce mémoire ne tiennent pas compte de tous les phénomènes de visibilité (tels que l'occlusion provoquée par l'action combinée de multiples bloqueurs), nous croyons que l'utilisation de tels bloqueurs et des méthodes présentées peuvent être étendues à d'autres types d'algorithmes et de phénomènes comme par exemple lors de parcours en temps réel de scène complexe, le rendu d'une image de synthèse.

Chapitre 7

Glossaire

Ce glossaire donne la définition utilisée dans le présent document pour certains termes. Comme l'utilisation du français en infographie peut parfois porter à confusion, les termes traduits ou empruntés de l'anglais se retrouvent ici.

agrégat *Cluster*. Groupe d'éléments (triangles, polygones, surfaces, etc.).

agrégat de faces *Face Cluster*. Groupe de polygones formant une région continue (partageant au moins un segment) englobée dans un volume bien défini.

agrégat de volumes *Volume Cluster*. Groupe de polygones englobé dans un volume englobant.

arbre BSP *Binary Space Partition Tree*. Décomposition hiérarchique d'un espace tridimensionnel telle que chaque région est divisée en deux à l'aide d'un plan.

arbre k-d *K-d tree*. Arbre de dimension k . Décomposition hiérarchique d'un espace de dimension k telle que chaque région est divisée en deux parties à l'aide d'un hyperplan aligné sur les axes.

arbre quaternaire *Quadtree*. Décomposition hiérarchique d'un espace bidimensionnel telle que chaque région est divisée en quatre quadrants égaux à l'aide de deux droites alignées sur les axes.

arbre octaire *Octree*. Décomposition hiérarchique d'un espace tridimensionnel telle que chaque région est divisée en huit octants égaux à l'aide de trois plans alignés sur les axes.

bloqué *Occludee*. Surface bloquée par un ou plusieurs bloqueurs.

bloqueur *Occluder*. Surface bloquant une région de l'espace par rapport à un point de vue ou encore par rapport à une autre région de l'espace.

BRDF *Bidirectional Reflectance Distribution Function*. Fonction hémisphérique à quatre dimensions (et plus) de réflectance d'une surface.

carte d'ombre *Shadow Map*. Image des profondeurs construite pour une lumière ponctuelle généralement à l'aide de l'algorithme du tampon de profondeur pour calculer la visibilité de la lumière.

cône de normales *cone of normals*. Cône formé d'une normale moyenne et d'un angle d'ouverture représentant l'ensemble des normales d'une surface.

élimination de surfaces par pyramide de vue *Frustrum Culling*. Élimination des surfaces d'une scène qui sont hors de la pyramide de vue.

élimination de surfaces par occlusion *Occlusion Culling*. Élimination des surfaces cachées par des bloqueurs.

enveloppe convexe *Convex Hull*. Polygone (2D) ou polyèdre (3D) convexe minimal englobant un polygone (2D) ou une surface polygonale (3D), respectivement.

facteur de forme *Form Factor*. Relation géométrique entre deux surfaces.

illumination directe Aussi appelée illumination locale. Illumination d'une surface provenant directement d'une source de lumière sans interaction avec une autre surface.

illumination globale Illumination tenant compte des interréllections entre la lumière et les surfaces.

illumination indirecte Partie de l'illumination globale sans l'illumination directe.

lancer de rayons *Ray Casting*. Technique de visibilité servant à détecter quelle surface est intersectée à partir d'un point et d'une direction (ou d'un autre point).

LDI *Layered Depth Images*. Tampons de profondeur et d'images formés en multicouches.

lustré *glossy*. Caractéristique d'une surface qui est semi-réfléctive.

maillage *Mesh*. Définition de la topologie d'une surface généralement par l'interconnexion de polygones.

modeleur *Modeler*. Logiciel de modélisation de surfaces tridimensionnelles.

occlusion Le terme le plus approprié serait occultation, mais en anglais comme en français, occlusion est plus couramment utilisé. Il désigne le blocage d'une région ou d'un certain nombre de surfaces par un ou plusieurs bloqueurs.

oracle Section de l'algorithme de radiosit  hi rarchique qui d termine   quel niveau l' change entre deux  l ments doit avoir lieu.

parcours *Walkthrough*. Chemin libre suivi par un usager (une cam ra) dans un environnement synth tique.

photor aliste Ce dit d'une image de synth se lorsque celle-ci atteint un niveau de r alisme presque indissociable d'une photographie de la r alit .

portal Ouverture au travers de laquelle la visibilit  se propage.

PVS *Potentially Visible Set*. Ensemble de cellules pour lesquelles l'ensemble des surfaces potentiellement visibles   partir de celle-ci est calcul .

pyramide de visibilit  Pyramide d finissant le demi-espace pour lequel un bloqueur directionnel est valide.

radiosit  Technique de rendu photor aliste par  l ments finis tenant en compte l'interr flexion de la lumi re sur les surfaces.

Se dit aussi de la puissance lumineuse totale  mise par unit  de surface en un point. Les unit s sont le ($\frac{\text{watt}}{\text{m}^2}$).

radiosit  hi rarchique par agr gats *Hierarchical Cluster Radiosity*. Technique de radiosit  dans laquelle on regroupe les surfaces de la sc ne en agr gats afin d'acc l rer les  changes d' nergie lumineuse.

rayon d'ombre *Shadow Ray*. Rayon retournant une valeur comprise entre [0,1] indiquant le degr  d'occlusion entre deux points.

rendu *Rendering*. Processus de fabrication d'une image de synth se.

rendu par balayage de lignes *Scanline rendering*. Algorithme de rendu d composant les primitives en lignes (g n ralement horizontales) afin de d terminer celles visibles du point de vue de la cam ra.

tampon de profondeur *Z-Buffer*. Tampon dans lequel pour chaque pixel d'une image est associ e sa profondeur en son centre.

trac  de rayons *Ray Tracing*. Algorithme de rendu utilisant le lancer de rayons r cursif pour tenir compte des diff rents effets de lumi re telles la r flexion et

la réfraction.

Bibliographie

- [AH93] L. Aupperle et P. Hanrahan. « A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection ». *Computer Graphics*, volume 27, pages 155–162, 1993.
- [Air90] J. M. Airey. *Increasing Update Rates in the Building Walkthrough System with Automatic Model-Space Subdivision and Potentially Visible Set Calculations*. Thèse de doctorat, Dept. of CS, U. of North Carolina, juillet 1990.
- [App68] A. Appel. « Some Techniques for Shading Machine Renderings of Solids ». Dans *AFIPS 1968 Spring Joint Computer Conf.*, volume 32, pages 37–45, 1968.
- [ARHM00] M. Agrawala, R. Ramamoorthi, A. Heirich et L. Moll. « Efficient Image-Based Methods for Rendering Soft Shadows ». *Proceedings of SIGGRAPH 2000*, pages 375–384, juillet 2000.
- [ASVN00] C. Andújar, C. Saona-Vázquez et I. Navazo. « LOD Visibility Culling and Occluder Synthesis ». *Computer Aided Design*, 2000. Accepted for publication.
- [AW87] J. Amanatides et A. Woo. « A fast voxel traversal algorithm for ray tracing ». Dans *Eurographics '87*, pages 3–10, août 1987.
- [BMSW91] D. R. Baum, S. Mann, K. P. Smith et J. M. Winget. « Making radiosity usable : Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions ». Dans *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 51–60, juillet 1991.

- [Bou70] W. J. Bouknight. « A Procedure for Generation of Three-Dimensional Half-Toned Computer Graphics Presentations ». *Communications of the ACM*, volume 13, numéro 9, pages 527–536, septembre 1970.
- [BRW89] D. R. Baum, H. E. Rushmeier et J. M. Winget. « Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors ». Dans *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 325–334, juillet 1989.
- [Cat74] E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Thèse de doctorat, University of Utah, décembre 1974.
- [CCOL98] Y. Chrysanthou, D. Cohen-Or et D. Lischinski. « Fast Approximate Quantitative Visibility for Complex Scenes ». Dans *Proceedings of the Conference on Computer Graphics International 1998 (CGI-98)*, pages 220–227, juin 22–26 1998.
- [CCWG88] M. F. Cohen, S. E. Chen, J. R. Wallace et D. P. Greenberg. « A Progressive Refinement Approach to Fast Radiosity Image Generation ». Dans *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, août 1988.
- [CG85] M. F. Cohen et D. P. Greenberg. « The Hemi-Cube : A Radiosity For Complex Environments ». Dans *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 31–40, juillet 1985.
- [CGIB86] M. F. Cohen, D. P. Greenberg, D. S. Immel et P. J. Brock. « An Efficient Radiosity Approach for Realistic Image Synthesis ». *IEEE Computer Graphics and Applications*, volume 6, numéro 3, pages 26–35, mars 1986.
- [CMS98] P. Cignoni, C. Montani et R. Scopigno. « A comparison of mesh simplification algorithms ». *Computers & Graphics*, volume 22, numéro 1, pages 37–54, février 1998.
- [COFHZ98] D. Cohen-Or, G. Fibich, D. Halperin et E. Zadicario. « Conservative Visibility and Strong Occlusion for Viewspace Partitioning of Densely Occluded Scenes ». *Computer Graphics Forum (Eurographics '98)*, volume 17, numéro 3, pages 243–254, 1998.

- [CPC84] R. L. Cook, T. Porter et L. Carpenter. « Distributed Ray Tracing ». Dans *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 137–145, juillet 1984.
- [CT97] S. Coorg et S. Teller. « Real-Time Occlusion Culling for Models with Large Occluders ». Dans *1997 Symposium on Interactive 3D Graphics*, pages 83–90, avril 1997.
- [CVM⁺96] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. P. Brooks, Jr. et W. Wright. « Simplification Envelopes ». Dans *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 119–128, août 1996.
- [CW93a] Shenchang Eric Chen et Lance Williams. « View Interpolation for Image Synthesis ». Dans *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 279–288, août 1993.
- [CW93b] M. F. Cohen et J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, 1993.
- [DDP97] F. Durand, G. Drettakis et C. Puech. « The Visibility Skeleton : A Powerful and Efficient Multi-Purpose Global Visibility Tool ». *Proceedings of SIGGRAPH '97*, pages 89–100, août 1997.
- [DDP99] F. Durand, G. Drettakis et C. Puech. « Fast and Accurate Hierarchical Radiosity Using Global Visibility ». *ACM Transactions on Graphics*, volume 18, numéro 2, pages 128–170, avril 1999.
- [DDTP00] F. Durand, G. Drettakis, J. Thollot et C. Puech. « Conservative Visibility Preprocessing Using Extended Projections ». *Proceedings of SIGGRAPH 2000*, pages 239–248, juillet 2000.
- [DF94] G. Drettakkis et E. Fiume. « A Fast Shadow Algorithm for Area Light Sources Using Backprojection ». Dans *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 223–230, juillet 1994.
- [Dur99] F. Durand. *3D Visibility, Analysis and Applications*. Thèse de doctorat, Université Joseph Fourier, Grenoble, 1999.

- [Ede87] H. Edelsbrunner. « Algorithms in Combinatorial Geometry ». Dans *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.
- [EST⁺95] B. L. Evans, C. Schwarz, J. Teich, A. Vainshtein et E. Welzl. « Minimal Enclosing Parallelogram with Application ». Dans *Proceedings of the 11th Annual Symposium on Computational Geometry*, pages C34–C35, juin 1995.
- [FK85] K. Fujimura et T. L. Kunii. « A Hierarchical space indexing method ». Dans *Computer Graphics Visual Technology and Art (Proceedings of Computer Graphics Tokyo '85)*, pages 21–33. Springer-Verlag, 1985.
- [FKN80] H. Fuchs, Z. M. Kedem et B. F. Naylor. « On Visible Surface Generation by a Priori Tree Structures ». Dans *Computer Graphics (SIGGRAPH '80 Proceedings)*, volume 14, pages 124–133, juillet 1980.
- [FvFH90] J. Foley, A. van Dam, S. Feiner et J. Hughes. *Computer graphics : Principles and Practice*. Addison-Wesley, 2e édition, 1990.
- [Gar99] M. Garland. *Quadric-Based Polygonal Surface Simplification*. Thèse de doctorat, Carnegie Mellon University, 1999.
- [GDW00] X. Granier, G. Drettakis et B. Walter. « Fast Global Illumination Including Specular Effects ». Dans *Workshop on Rendering 2000*, pages 200–212, New York, NY, juin 2000. Springer Wien.
- [GH96] S. Gibson et R. J. Hubbard. « Efficient Hierarchical Refinement and Clustering for Radiosity in Complex Environments ». *Computer Graphics Forum*, volume 15, numéro 5, pages 297–310, 1996.
- [GK93] N. Greene et M. Kass. « Hierarchical Z-Buffer Visibility ». Dans *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 231–240, 1993.
- [Gla84] A. S. Glassner. « Space sub-division for fast ray tracing ». *IEEE Computer Graphics and Applications, October 1984*, volume 4, pages 15–22, 1984.
- [Gla89] A. Glassner. *An Introduction to Ray Tracing*. Academic Press, 1989.
- [Gou71] H. Gouraud. « Continuous Shading of Curved Surfaces ». *IEEE Transactions on Computers*, volume C-20, numéro 6, pages 623–629, juin 1971.

- [Gra92] C. W. Grant. *Visibility Algorithms in Image Synthesis*. Thèse de doctorat, U. of California, Davis, 1992.
- [GS87] J. Goldsmith et J. Salmon. « Automatic Creation of Object Hierarchies for Ray Tracing ». *IEEE Computer Graphics & Applications*, volume 7, numéro 5, pages 14–20, mai 1987.
- [GTGB84] C. M. Goral, K. K. Torrance, D. P. Greenberg et B. Battaile. « Modeling the Interaction of Light Between Diffuse Surfaces ». Dans *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 213–222, juillet 1984.
- [HDG99] D. Hart, P. Dutré et D. P. Greenberg. « Direct Illumination With Lazy Visibility Evaluation ». Dans *Proceedings of SIGGRAPH '99*, pages 147–154, août 8–13 1999.
- [HG97] P. S. Heckbert et M. Garland. « Survey of Polygonal Surface Simplification Algorithms ». Rapport technique, CS Dept., Carnegie Mellon U., 1997.
- [HMC⁺97] T. Hudson, D. Manocha, J. Cohen, M. Lin, K. Hoff et H. Zhang. « Accelerated Occlusion Culling Using Shadow Frustra ». Dans *Proceedings of the Thirteenth Annual Symposium on Computational Geometry*, pages 1–10, juin 1997.
- [HS67] H. C. Hottel et A. F. Sarofim. *Radiative Transfer*. McGraw Hill, New York, NY, 1967.
- [HSA91] P. Hanrahan, D. Salzman et L. Aupperle. « A Rapid Hierarchical Radiosity Algorithm ». Dans *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, juillet 1991.
- [HW91] E. Haines et J. Wallace. « Shaft Culling for Efficient Ray-Traced Radiosity ». Dans *Eurographics Workshop on Rendering*, pages 122–138, 1991.
- [ICG86] D. S. Immel, M. F. Cohen et D. P. Greenberg. « A Radiosity Method for Non-diffuse Environments ». Dans *Proceedings of SIGGRAPH '86*, volume 20, pages 133–142, 1986.
- [Jen96] H. W. Jensen. « Global Illumination using Photon Maps ». Dans *Eurographics Rendering Workshop 1996*, pages 21–30, juin 1996.

- [Jol86] I. T. Jolliffe. *Principal Component Analysis*. Series in Statistics. Springer-Verlag, 1986.
- [Kaj86] J. T. Kajiya. « The Rendering Equation ». Dans *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, août 1986.
- [KCCO00] V. Koltun, Y. Chrysanthou et D. Cohen-Or. « Virtual Occluders : An efficient Intermediate PVS representation ». *11th Eurographics Workshop on Rendering*, pages 59–70, juin 2000.
- [LP00] L. Leblanc et P. Poulin. « Guaranteed Occlusion and Visibility in Cluster Hierarchical Radiosity ». *11th Eurographics Workshop on Rendering*, pages 89–100, juin 2000.
- [LT99] F.-A. Law et T.-S. Tan. « Preprocessing Occlusion for Real-Time Selective Refinement ». Dans *1999 Symposium on Interactive 3D Graphics*, pages 47–52, 1999.
- [LTG93] D. Lischinski, F. Tampieri et D. P. Greenberg. « Combining Hierarchical Radiosity and Discontinuity Meshing ». *Proceedings of SIGGRAPH '93*, pages 199–208, août 1993.
- [Lue97] D. Luebke. « A Survey of Polygonal Simplification Algorithms ». Rapport technique TR97-045, Department of Computer Science, University of North Carolina - Chapel Hill, décembre 1997.
- [LW93] E. P. Lafortune et Y. D. Willems. « Bi-directional Path Tracing ». Dans H. P. Santo, éditeur. *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, décembre 1993.
- [Max89] N. Max. « Smooth appearance for polygonal surfaces ». *The Visual Computer*, volume 5, numéro 3, pages 160–173, juin 1989.
- [Men98] D. Meneveaux. *Lighting Simulation in Complex Architectural Environments : sequential and parallel approaches*. Thèse de doctorat, Université de Rennes I, 1998.
- [MSF99] G. Mueller, S. Schaefer et D. Fellner. « Automatic Creation of Object Hierarchies for Radiosity Clustering ». *Pacific Graphics '99*, octobre 1999.

- [MWCF90] J. Marks, R. Walsh, J. Christensen et M. Friedell. « Image and Intervisibility Coherence in Rendering ». Dans *Proceedings of Graphics Interface '90*, pages 17–30, mai 1990.
- [PD90] H. Plantinga et C. R. Dyer. « Visibility, Occlusion, and the Aspect Graph ». *International Journal of Computer Vision*, volume 5, numéro 2, pages 137–160, 1990.
- [Pla93] H. Plantinga. « Conservative Visibility Preprocessing for Efficient Walkthroughs of 3D Scenes ». Dans *Proceedings of Graphics Interface '93*, pages 166–173, mai 1993.
- [Pue91] X. Pueyo. « Diffuse Interreflections. Techniques for Form-Factor Computation : A Survey ». *The Visual Computer*, volume 7, numéro 4, pages 200–209, juillet 1991.
- [Rei92] M. C. Reichert. « A Two-Pass Radiosity Method Driven by Lights and Viewer Position ». Mémoire de maîtrise, Cornell University, janvier 1992.
- [RT90] H. E. Rushmeier et K. E. Torrance. « Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials ». *ACM Transactions on Graphics*, volume 9, numéro 1, pages 1–27, janvier 1990.
- [RW80] S. M. Rubin et T. Whitted. « A 3-Dimensional Representation for Fast Rendering of Complex Scenes ». Dans *Computer Graphics (SIGGRAPH '80 Proceedings)*, volume 14, pages 110–116, juillet 1980.
- [SA93] L. A. Shirman et S. S. Abi-Ezzi. « The Cone of Normals Technique for Fast Processing of Curved Patches ». Dans *Eurographics '93*, pages 261–272, 1993.
- [SAG94] B. Smits, J. Arvo et D. Greenberg. « A Clustering Algorithm for Radiosity in Complex Environments ». Dans *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442, juillet 1994.
- [SDDS00] G. Schaufler, J. Dorsey, X. Decoret et F. X. Sillion. « Conservative Volumetric Visibility with Occluder Fusion ». *Proceedings of SIGGRAPH 2000*, pages 229–238, juillet 2000.

- [SDS95] F. Sillion, G. Drettakis et C. Soler. « A Clustering Algorithm for Radiance Calculation in General Environments ». Dans *Eurographics Rendering Workshop 1995*, pages 196–205, juin 1995.
- [SG94] A. J. Stewart et S. Ghali. « Fast Computation of Shadow Boundaries Using Spatial Coherence and Backprojections ». Dans *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 231–238, juillet 1994.
- [SGHS98] J. W. Shade, S. J. Gortler, Li-wei He et R. Szeliski. « Layered Depth Images ». Dans *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 231–242, juillet 1998.
- [SH93] P. Schroeder et P. Hanrahan. « A Closed Form Expression for the Form Factor between Two Polygons ». Technical Report TR-404-93, Princeton University, Computer Science Department, janvier 1993.
- [She88] M. S. Shephard. « Approaches to the automatic generation and control of finite element meshes ». *Applied Mechanics Review*, volume 41, pages 169–185, 1988.
- [Sil94] F. Sillion. « Clustering and Volume Scattering for Hierarchical Radiosity Calculations ». Dans *Fifth Eurographics Workshop on Rendering*, pages 105–117, juin 1994.
- [Sil95] F. X. Sillion. « A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters ». *IEEE Transactions on Visualization and Computer Graphics*, volume 1, numéro 3, pages 240–254, septembre 1995.
- [Smi98] B. Smits. « Efficiency Issues for Ray Tracing ». *Journal of Graphics Tools*, volume 3, numéro 2, pages 1–14, 1998.
- [SP94] F. Sillion et C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, 1994.
- [SS98] C. Soler et F. X. Sillion. « Fast Calculation of Soft Shadow Textures Using Convolution ». Dans *Proceedings of SIGGRAPH '98*, Annual Conference Series, pages 321–332, juillet 1998.

- [SSS74] I. E. Sutherland, R. F. Sproull et R. A. Schumacker. « A Characterization of Ten Hidden-Surface Algorithms ». *ACM Computing Surveys*, volume 6, numéro 1, pages 1–55, mars 1974.
- [SSSS98] M. Stamminger, H. Schirmacher, P. Slusallek et H.-P. Seidel. « Getting Rid of Links in Hierarchical Radiosity ». *Computer Graphics Forum*, volume 17, numéro 3, pages 165–174, 1998.
- [SVNB99] C. Saona-Vásquez, I. Navazo et P. Brunet. « The visibility octree : a data structure for 3D navigation ». *Computers & Graphics*, volume 23, numéro 5, pages 635–643, octobre 1999.
- [TH93] S. Teller et P. Hanrahan. « Global Visibility Algorithms for Illumination Computations ». Dans *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 239–246, 1993.
- [TS91] S. J. Teller et C. H. Séquin. « Visibility Preprocessing for Interactive Walkthroughs ». Dans *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 61–69, juillet 1991.
- [Tur90] G. Turk. « Generating Random Points in Triangles ». *Graphics Gems*, pages 24–28, 1990.
- [VG97] E. Veach et L. J. Guibas. « Metropolis Light Transport ». Dans *Proceedings of SIGGRAPH '97, Annual Conference Series*, pages 65–76, août 1997.
- [WA77] K. Weiler et K. Atherton. « Hidden surface removal using polygon area sorting ». *Computer Graphics (Proceedings of SIGGRAPH 77)*, volume 11, numéro 2, pages 214–222, juillet 1977.
- [War69] J. Warnock. « A Hidden-Surface Algorithm for Computer Generated Half-Tone Pictures ». Rapport technique TR 4–15, NTIS AD-733 671, University of Utah, Computer Science Department, 1969.
- [WBP98] Y. Wang, H. Bao et Q. Peng. « Accelerated Walkthroughs of Virtual Environments based on Visibility Processing and Simplification ». *Computer Graphics Forum (Eurographics '98)*, volume 17, numéro 3, pages 187–194, 1998.
- [WEH89] J. R. Wallace, K. A. Elmquist et E. A. Haines. « A Ray Tracing Algorithm for Progressive Radiosity ». Dans *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 315–324, juillet 1989.

- [WHG99] A. Willmott, P. Heckbert et M. Garland. « Face Cluster Radiosity ». Dans *Eurographics Workshop on Rendering*, pages 293–304, juin 1999.
- [Whi80] T. Whitted. « An improved illumination model for shaded display ». *CACM*, volume 23, numéro 6, pages 343–349, 1980.
- [Wil78] L. Williams. « Casting curved shadows on curved surfaces ». Dans *Computer Graphics (SIGGRAPH '78 Proceedings)*, volume 12, pages 270–274, août 1978.
- [WPF90] A. Woo, P. Poulin et A. Fournier. « A Survey of Shadow Algorithms ». *IEEE Computer Graphics and Applications*, volume 10, numéro 6, pages 13–32, novembre 1990.
- [WWS00] P. Wonka, M. Wimmer et D. Schmalstieg. « Visibility Preprocessing with Occluder Fusion for Urban Walkthroughs ». *11th Eurographics Workshop on Rendering*, pages 71–82, juin 2000.
- [Yam26] Z. Yamauti. « The Light Flux Distribution of a System of Interreflecting Surfaces ». *Journal of the Optical Society of America*, volume 13, pages 561–571, novembre 1926.
- [ZMHH97] H. Zhang, D. Manocha, T. Hudson et K. E. Hoff III. « Visibility Culling Using Hierarchical Occlusion Maps ». Dans *Proceedings of SIGGRAPH '97*, Annual Conference Series, pages 77–88, août 1997.