



HAL
open science

De l'analyse à la conception d'algorithmes pour une radiosité hiérarchique efficace

Jérémie Turbet

► **To cite this version:**

Jérémie Turbet. De l'analyse à la conception d'algorithmes pour une radiosité hiérarchique efficace. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2002. Français. NNT : . tel-00528811

HAL Id: tel-00528811

<https://theses.hal.science/tel-00528811>

Submitted on 22 Oct 2010

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

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

De l'analyse à la conception d'algorithmes pour une radiosité hiérarchique efficace

Jérémie TURBET

Thèse présentée pour l'obtention du titre de Docteur de l'Université Joseph Fourier
Spécialité Informatique
Arrêté ministériel du 5 juillet 1984 et du 30 mars 1992
Préparée au sein du laboratoire *iMAGIS-GRAVIR/IMAG-INRIA*. UMR CNRS C5527.
Soutenance prévue au printemps 2002

Composition du jury :

Hans-Peter	SEIDEL	Rapporteur
Bernard	PEROCHE	Rapporteur
Neil	GATENBY	Examineur
Claude	PUECH	Examineur
François	SILLION	Directeur de thèse

Remerciements

Cette thèse, mon bébé enfanté dans la douleur, est le fruit de l'ensemble des personnes qui m'ont entouré. Je tiens à remercier en tout premier lieu mes parents qui m'ont toujours soutenu sur la voie qui a guidé mes pas, malgré les chemins tortueux que j'ai pu emprunter.

Sur le plan humain, la dédicace suivante est pour jojo, colocataire et amie sans défauts (sans trop, en tous cas ...), avec qui j'ai beaucoup enrichi mon expérience. Gillou et jojo, "grâce" à leur statut de colloc', furent les principaux relecteurs patients et émérites de mon style inné de l'écriture fait de phrases à rallonges où les points et les virgules se phagocytent en un flou artistique, somme toute et à première vue, difficile à comprendre lors de la première relecture.

Sur le plan scientifique, je ne pourrais jamais assez remercier François –le vénérable– pour m'avoir offert la possibilité de réaliser ce travail, pour ses conseils toujours justes, pour son agréable personnalité et pour le temps qu'il a su m'accorder au bon moment malgré son emploi du temps plus que surchargé.

J'offre une mention spéciale à notre logiciel de développement : *Bright*. En dépit de tout ce que l'on peut en dire, il a au moins un avantage : celui d'obliger ceux qui travaillent dessus à se voir régulièrement afin de comprendre les méandres de sa structure interne (nommés aussi hack).

Un petit meuh au channel IRC #image pour leur présence virtuelle continue et particulièrement à NaD et Bounet pour avoir rendu cette machine rébarbative qu'est *nigelle* un peu plus vivante.

Il serait hérétique d'oublier le *barberousse* et le *basse-terre* sources d'inspiration et d'expiation durant ces trois années. Merci aussi à Amandine pour avoir rééduqué, torturé, massé, électrocuté et remusclé ce genou pendant plus de deux ans sans trop me lasser – un exploit –.

De manière plus éloignée je voudrais gratifier les amis et professeurs qui ont créé l'intérêt que je porte à l'imagerie de synthèse : soit Bruno Levy pour m'avoir fait découvrir le principe du ray-tracing en terminale ; puis le méro pour avec qui nous avons auto-développé notre motivation (et un peu plus). Merci surtout à Claude Secroun pour avoir créé le DUIIC et à Bernard Peroche qui m'ont permis de faire *enfin* ce que j'aimais.

Plus généralement et parce que je ne peux citer toutes les personnes que je voudrais, je tiens à remercier du fond du cœur tous ceux qui m'ont permis d'arriver à la réalisation de ce travail, ainsi que ceux qui ont supporté les effets de bord.

SOMMAIRE

Introduction	1
1 Le rendu en synthèse d'image	5
2 L'algorithme de radiosit�	25
3 Avatars de surfaces	41
4 Le raffinement	69
5 Visibilit�	97
6 Reconstruction	127
Conclusion	139
Pistes de recherche	145
A Transformation espace sph�rique - espace cart�sien	149
Table des mati�res	153
Table des figures	157
Bibliographie	165

Introduction

Le peintre qui s'apprête à peindre le soleil fait des théories, et, quand il veut commencer, le soleil n'est plus là.

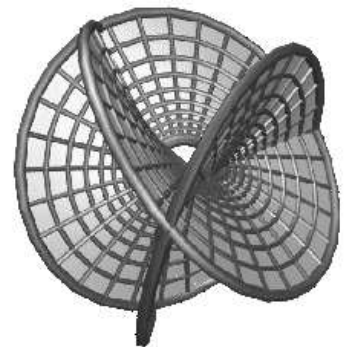
Jules Renard

L'imagerie de synthèse cherche à produire toutes sortes d'images numériques. Les images peuvent être générées à partir de rien par un graphiste comme un peintre devant sa toile. Elles peuvent prendre leur source dans une description mathématique ou géométrique pour pouvoir visualiser des fonctions complexes (visualisation scientifique, génération de texture) ou des environnements (rendu). Elles peuvent aussi être la composition d'autres images (rendu à base d'images, morphing).

La création des images nécessite en amont une description des données à visualiser. L'image ne peut pas exister sans cette description et les données brutes sont difficilement exploitables sans visualisation. Cette interdépendance fait que l'imagerie de synthèse n'est pas uniquement un travail de visualisation mais aussi un travail de génération de données (simulation, modèles physiques, animation).

Le rendu est la création d'une image à partir d'une description géométrique d'un environnement. Les premières recherches se sont naturellement tournées vers la création d'images "réalistes" en essayant de copier l'appareil photo puis l'oeil (rendu photo-réaliste) et en essayant de simuler les lois de l'optique (illumination globale).

Actuellement on cherche aussi à créer des images simplifiées dans leur aspect à partir de ces mêmes descriptions géométriques (rendu non photo-réaliste).



Illumination globale.

L'illumination globale est le domaine du rendu qui essaye de produire des images d'une qualité qui se veut proche de la réalité grâce à l'étude de la lumière et de toutes ses interactions avec son environnement. Actuellement deux grandes familles d'algorithmes permettent la création d'images "réalistes", l'une basée sur une évaluation ponctuelle des échanges lumineux : les algorithmes stochastiques de suivi de rayons ; l'autre basée sur une évaluation surfacique : les algorithmes de radiosité.

Les deux méthodes ne résolvent pas exactement le même type d'éclairage, elles sont même complémentaires, mais les essais de combinaison des deux ne sont pas encore satisfaisants. Les algorithmes stochastiques sont particulièrement performants dans des environnements très réfléchissants ou brillants, les algorithmes de radiosité ne gèrent que les environnements diffus, mats.

L'intérêt principal de l'illumination globale réside dans l'éclairage indirect, c'est-à-dire éclairage qui ne provient pas directement des sources lumineuses mais de réflexions multiples sur des surfaces.

Applications de la synthèse d'images.

Le rendu et l'illumination globale sont les domaines de l'imagerie de synthèse qui la rendent particulièrement propice à la création d'effets spéciaux ou de films complets. Les dernières années ont vu la naissance et la croissance de tels films, utilisant le rendu photo-réaliste pour l'intégration d'images numériques au sein d'images filmées ou le rendu non photo-réaliste pour des dessins d'animation.



Jurassic park © Industrial Light and Magic

Quelques logiciels professionnels utilisent les deux types d'algorithmes d'illumination globale et permettent aux graphistes de créer des images qu'il est difficile de différencier des images "naturelles".

Au vu de la qualité des images produites par l'utilisation des algorithmes actuels et les images produites dans la recherche, il est légitime de se poser des questions sur l'utilité de ces recherches. Notamment, certains effets, tels que le rendu d'objets de dimension très réduite (fourrure, herbe, fumée, etc.) ou la simulation de vêtements et drapés sont utilisés de façon convaincante pendant que les recherches sur le domaine sont loin d'être abouties.

La réalisation d'effets encore mal maîtrisés utilise la combinaison d'une méthode brutale mais robuste, d'une puissance de calcul très importante et d'une longue phase de retouche manuelle. Ceci à un coût de réalisation en temps et en argent important.

Dernièrement (2001), l'équipe de Pixar a créé un personnage de synthèse (Sullivan) couvert d'une fourrure longue et duveteuse. Pour obtenir une fourrure réaliste, ils ont modélisé 2,3 millions de poils en leur associant à chacun une géométrie. L'animation a été créée en appliquant une simulation par modèle physique à chacune des primitives. La présence du personnage dans le film représente près de 100.000 images. Une armée de graphistes et une puissance de calcul impressionnante permettent de donner un résultat extrêmement convaincant.



Monsters, Inc. © Pixar

Dans le domaine de l'illumination globale, avant que les algorithmes actuels sachent évaluer correctement cet éclairage, des images commerciales permettaient de le visualiser... Le travail minutieux des graphistes était de placer un ensemble de sources de lumière – ou de sources d'ombres – dans la scène pour recréer artificiellement l'éclairage indirect.

L'apport de nouvelles technologies est recherché par les graphistes afin de réduire leur charge de travail en automatisant ce que la perception humaine caractérise inconsciemment. La combinaison de la recherche et de l'application massive et immédiate des résultats (sans oublier l'intérêt commercial !) produit un développement rapide de l'imagerie de synthèse.

Dans ce cadre, ce mémoire traite de la partie de l'illumination globale appliquée aux surfaces diffuses : la *radiosité*.

La radiosité.

L'algorithme de radiosité permet de calculer l'éclairage direct et indirect dans une scène. Sa plus grosse hypothèse est que toutes les surfaces composant la scène sont mates. Il n'existe quasiment aucun endroit réel où ces conditions sont réunies. Toutefois cette limitation forte permet à l'éclairage d'être indépendant du point de vue. Il est alors possible, une fois l'éclairage calculé, de visiter interactivement la scène, ce qui est fortement utilisé en réalité virtuelle.

L'algorithme de radiosité existe depuis une vingtaine d'années. La théorie bien définie des échanges radiatifs sur laquelle il se base lui a conféré le statut de solution analytique au problème de l'illumination globale pour des surfaces diffuses. Mais la complexité des scènes empêche une résolution brutale analytique, il faut appliquer des méthodes de simplification de systèmes d'équations : la *radiosité hiérarchique*. La complexité de la radiosité hiérarchique étant encore trop élevée, la hiérarchie de surfaces a été étendue à une hiérarchie volumique : le *regroupement de surfaces*.



©iMAGIS

Chacune de ces améliorations a réduit le temps de calcul et a amélioré la convergence de la solution mais a aussi apporté une difficulté d'utilisation et des approximations supplémentaires. Ce qui donne aujourd'hui un algorithme complexe et peu stable.

Le principe de la radiosité hiérarchique est de calculer les échanges radiatifs à des niveaux de résolution des éléments bien choisis afin de limiter le nombre de calculs. Derrière le terme "bien choisis" se cache une heuristique très complexe de choix des niveaux. Sans une heuristique adéquate, l'intérêt de la radiosité hiérarchique est limité. Les nombreuses recherches n'ont pas abouti à une heuristique précise et définie.

Les nombreuses heuristiques étudiées se limitent à des cas particuliers et sont de nature hétérogène. Il est important, pour que la radiosité hiérarchique soit définitivement utilisable, qu'une heuristique correcte soit trouvée.

Contributions.

Dans cette thèse, centrée sur l’heuristique du raffinement, nous avons étudié les 3 différentes étapes de l’algorithme de radiosit  hi rarchique avec regroupement de surfaces. Nous avons aussi  tudi  le goulot d’ tranglement de l’algorithme : l’ valuation et le calcul du facteur de visibilit .

Le pr -traitement.

Le calcul de la solution est li    la g om trie de la sc ne o  un nombre important d’approximations sont faites. Les ensembles de petits polygones sont sources d’erreurs et de temps de calculs suppl mentaires. L’id e est de remplacer des ensembles de petits polygones par une g om trie r duite pour simplifier et acc l rer leur calcul : *les avatars de surfaces*. Nous proposons une  tude de construction automatique de ces avatars et concluons par d terminer une m thode optimale.

Le calcul de la solution.

Le calcul   proprement parler de la solution utilise l’heuristique de raffinement pour d terminer   quel niveau hi rarchique effectuer les  changes radiatifs. Notre  tude comporte deux points distincts : la cr ation d’une structure du raffinement sous forme de graphe qui permet de d couper le complexe probl me du raffinement en briques  l mentaires ; puis la r alisation d’un raffineur utilisant cette structure.

Le post-traitement.

Une fois la solution calcul e chaque  l ment hi rarchique poss de une couleur. L’ tape permettant de passer de cette solution   une image ou une solution compl te est la reconstruction de la fonction. Nous proposerons une fonction de reconstruction permettant d’ liminer les art facts engendr s par les m thodes classiques de reconstruction.

La visibilit .

L’ valuation de la fonction de radiosit  utilise le calcul d’un noyau d’ change radiatif : le facteur de forme ; ce facteur inclus un calcul de visibilit  entre deux  l ments qui est difficile et long   estimer. La visibilit  est un domaine de recherche important car probl matique dans beaucoup d’algorithmes.

Diff rents outils permettent l’ valuation du facteur de visibilit  ou de la classification de celle-ci. Nous proposons une am lioration de l’outil volumique de classification : les *volumes de vision* ; en lui ajoutant la d termination exacte de certains cas d’invisibilit  totale.

Les outils de classification par  chantillonnage n cessitent un nombre de calculs important d’intersections. Nous proposons une optimisation bas e sur un parcours du graphe de visibilit    partir du dernier obstacle intersect , afin de tirer parti de la coh rence spatiale des obstacles.

Enfin, nous proposons une  tude statistique (non limit e   la radiosit ) ainsi qu’une fonction de pr diction de la classification de la visibilit  entre deux  l ments en fonction des obstacles. Nous terminerons par une applications de ces statistiques dans des sc nes de radiosit .

Le rendu en synthèse d'image

L'espace est noir parce que la lumière des étoiles ne trouve pas de paroi pour se refléter. Alors les rayons de lumière s'épuisent dans l'infini. Le jour où l'on verra une légère couleur dans le fond de l'univers, c'est que nous aurons atteint l'un de ses coins.

Bernard Werber

A fin de satisfaire un besoin grandissant de réalisme poussé par la réalisation d'effets spéciaux, de jeux vidéo, de simulations et d'imagerie médicale, la recherche dans le domaine de la création d'images réalistes connaît un essor important. Dans ce chapitre, nous allons nous intéresser à la génération d'images de synthèse photoréalistes, et particulièrement à la simulation d'éclairage. Le principe de base est de suivre tous les chemins que la lumière peut prendre entre deux points : un récepteur (oeil, appareil photographique) et un émetteur (lampe, source lumineuse). Pour cela nous parcourons une vaste échelle de grandeurs : nous partirons de la composition de la lumière, en étudiant les phénomènes physiques au niveau atomique puis nous verrons l'ensemble des simplifications faites afin de revenir à une échelle calculable. Les équations qui régissent l'ensemble des échanges lumineux sont connues, mais d'une complexité telle que les puissances de calcul actuelles sont très largement insuffisantes pour obtenir une solution précise. Dans une deuxième partie, nous verrons plusieurs méthodes pour approcher ces équations, chacune traitant plus ou moins bien certaines configurations.

1 Lumière et grandeurs lumineuses.

Les "images" que nous percevons de la réalité qui nous entoure se forment en plusieurs étapes. La première est l'ensemble des trajets de la lumière dans l'espace et le temps, comprenant des réflexions, réfractions, irisations, polarisations, etc... La seconde est la stimulation de notre oeil par ces trajets de lumière arrivant de toutes les directions à un instant donné. Ces stimuli excitent le système perceptif de notre cerveau pour former une "image". La compréhension de ces deux étapes est utile au but que nous recherchons : la création d'images réalistes. Les sections suivantes présentent un historique des découvertes sur la lumière, le système visuel humain et le formalisme scientifique pour étudier le domaine de la lumière.

1.1 Qu'est-ce que la lumière ?

La lumière est un phénomène étudié depuis longtemps, pourtant il est resté entièrement énigmatique jusqu'à il y a quelques années. La définition de F. Larousse en 1872 dans le *Nouveau dictionnaire de la langue française* montre la non-connaissance à son propos.

Lumière : n. f. Fluide subtil qui éclaire les objets et les rend visibles ; bougie, chandelle, lampe allumée.

Les définitions trouvées dans deux dictionnaires et encyclopédies actuels reflètent la dualité de sa représentation. La définition donnée par le *petit Robert* édition 1999 la définit exclusivement comme une onde électromagnétique :

Lumière : n. f. Ensemble des rayonnements électromagnétiques visibles, c'est-à-dire susceptibles d'être perçus directement par un oeil humain. Toutefois, on étend souvent le domaine de la lumière à des parties invisibles du spectre, l'infrarouge et l'ultraviolet, dont les propriétés restent très voisines de celles de la lumière "visible".

Alors que, à la même époque, le *Dictionnaire Universel Francophone* des éditions Hachette la définit principalement comme un flot de particules :

Lumière : n. f. PHYS Ensemble de particules élémentaires (nommées photons) se déplaçant à très grande vitesse (299 792,427 km/s dans le vide) et présentant les caractères d'une onde.

Nous allons voir avec un bref historique le pourquoi de ces deux définitions de la lumière.

Historique.

La nature de la lumière est restée très longtemps un mystère pour la science, elle fut découverte petit à petit aux XIX^e et XX^e siècles. Deux théories divisaient la communauté scientifique :

- Une théorie corpusculaire décrite par Isaac Newton en 1672 [Ron56] appuyée par sa fameuse "experimentum crucis". Par cette expérience, il montra la diffraction de la lumière blanche en un spectre lumineux. Cette Théorie fut immédiatement contestée par Robert Hooke et Christiaan Huygens qui lui opposaient une théorie proche de la théorie ondulatoire actuelle. La renommée de Newton était tellement importante que les "newtoniens" rejetèrent immédiatement les résultats des théories opposées.
- Une théorie ondulatoire dont les bases furent posées par Christiaan Huygens dans son "Traité de la lumière" écrit en 1678 et publié en 1691. Il considérait la lumière comme une onde se propageant dans un milieu mal défini qu'il appela *éther*. Puis Thomas Young relança cette théorie en 1804 en calculant les premières valeurs de longueurs d'ondes de la lumière, notamment celle du rouge (0,7 μm). Mais c'est grâce au mémoire d'Auguste Fresnel en 1818 et à la formulation des lois de l'électromagnétique par James-Clerk Maxwell en 1873 que la théorie ondulatoire fut quasi définitive et admise [Ron56].

Ces deux théories peuvent sembler contradictoires, la théorie ondulatoire se situant dans un domaine continu alors que la théorie corpusculaire est définitivement discontinue...

En 1905, Albert Einstein découvre l'effet photoélectrique qui permet de produire de l'énergie en utilisant la lumière ; il en fait une théorie corpusculaire vivement rejetée à cause de l'engouement de la communauté scientifique pour la théorie ondulatoire. Dans les années suivantes, ses travaux sont vérifiés expérimentalement. Il rédigea alors sa théorie de la mécanique quantique appliquée à la lumière où il décrit la décomposition des ondes électromagnétiques en deux composantes perpendiculaires (fig. 1.1). Dans cette théorie quantique, la lumière apparaît comme un flux discontinu de *photons* (ou *quanta*) qui sont des particules élémentaires de masse nulle au repos. L'énergie d'un quanta (E) est proportionnelle à la fréquence de l'onde.

$$E = h\nu \times n$$

$$\text{avec : } \begin{cases} h\nu & \text{constante de Planck} = 6.626075510^{-34} J.s \\ n & \text{fréquence de l'onde en Hertz} \end{cases}$$

La lumière n'est alors ni entièrement continue, ni entièrement discontinue, et ne se propage pas dans l'éther. Albert Einstein a reçu le prix Nobel de physique en 1921 pour ses nombreux travaux.

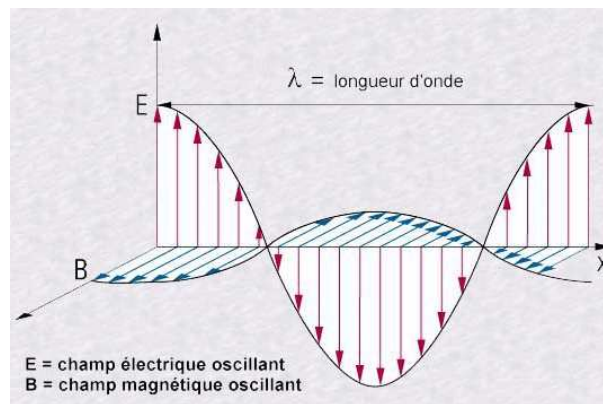


FIG. 1.1: Décomposition d'une onde électromagnétique en deux ondes perpendiculaires.

La définition actuelle de la lumière en tant que phénomène ondulatoire ou corpusculaire ne paraît pas suffisante. Cela semble une projection d'un processus complexe dans deux espaces différents, mais nous ne voyons que ses projections et non pas sa totalité.

La question reste en suspens : y a-t-il un nombre fini de projections jusqu'à la réalité, pourrions nous en avoir un jour une "vision" globale ?

1.2 L'oeil humain.

L'oeil humain est le récepteur sensible aux rayonnements électromagnétiques compris entre le violet et le rouge, extrémités du spectre lumineux (fig. 1.5) Les bornes définies par la Commission Internationale de l'Éclairage (CIE) sont 380 nm pour le violet et 780 nm pour le rouge [dl83], par contre certaines fonctions sont définies de $\sim 360 \text{ nm}$ à $\sim 830 \text{ nm}$, et l'usage courant en synthèse d'image utilise une plage de 400 nm à 700 nm .

Notre oeil est comparable à un appareil photographique où le cristallin correspond à la lentille, l'iris au diaphragme, la pupille à l'objectif, le vitré à la chambre noire, et la rétine à la pellicule (plan

focal). Le nerf optique transmet les informations reçues par l'oeil au cerveau (fig. 1.2). Comme dans un appareil photographique c'est la rétine qui est sensible aux stimuli lumineux, elle est tapissée de deux types de récepteurs : les cônes et les bâtonnets. Ils accaparent près de 70% des récepteurs du corps humain, leur répartition non uniforme dépend de l'angle par rapport au centre de la rétine (la fovéa) (fig. 1.3). Chacun d'entre eux possède des propriétés différentes.

- **Les cônes** : Il existe trois types de cônes qui diffèrent par la couleur qu'ils perçoivent. Ils sont sensibles au bleu, au vert ou au rouge. Les couleurs que nous voyons provoquent un mélange de ces trois sensations. Ils sont moins sensibles que les bâtonnets et ont besoin d'une intensité lumineuse plus importante pour être excités. La rétine d'un oeil humain possède 6 à 7 millions de cônes.
- **Les bâtonnets** : Ils sont 25 à 100 fois plus sensibles que les cônes ; ils nous servent à voir dans la pénombre. Ainsi, grâce à eux, nous pouvons voir malgré une très faible luminosité, mais nous ne percevons pas les couleurs car il n'y a qu'un seul type de bâtonnet. Un dicton nous le rappelle "La nuit, tous les chats sont gris". Nous le constatons aisément en observant de nuit un paysage éclairé par la pleine lune, il est impossible de distinguer les couleurs. Notre oeil possède 100 à 110 millions de bâtonnets.

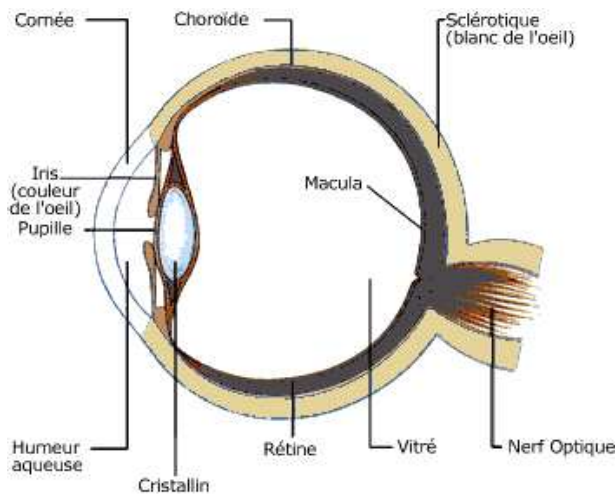


FIG. 1.2: Section d'un globe oculaire humain.

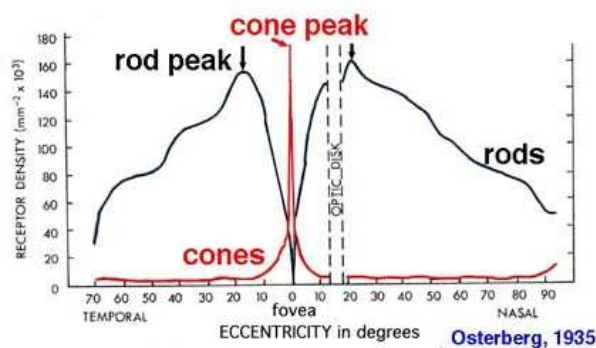


FIG. 1.3: Répartition des cônes et bâtonnets en fonction de l'angle à la fovéa.

Adaptation.

L'oeil est soumis à des luminances variant sur une dynamique importante de $\sim 10^{-5} \text{ cd.m}^{-2}$ (nuit noire) à $\sim 10^8 \text{ cd.m}^{-2}$ (plein soleil) *. L'oeil ne peut pas prendre en compte cette plage dans son ensemble, il s'adapte au niveau de luminance moyen. Cette adaptation prend un certain temps et n'est pas linéaire. Parce que nous ne traitons que des cas d'éclairage en régime constant, l'adaptation temporelle n'est pas présentée ici. La plage d'intensité lumineuse est habituellement découpée en 3 domaines :

- Domaine scotopique : radiance inférieure à $\sim 10^{-3} \text{ cd.m}^{-2}$. Intensité lumineuse très faible, par exemple en vision nocturne avec une faible lune. Dû à l'inefficacité des cônes à ces intensités, la résolution des détails est fortement diminuée et la perception des couleurs inexistante. Lorsque la luminosité est quasiment nulle, la vision est uniquement périphérique.
- Domaine mésopique : radiance comprise entre $\sim 10^{-3} \text{ cd.m}^{-2}$ et $\sim 10 \text{ cd.m}^{-2}$. Cette plage d'intensité est de l'ordre de grandeur d'un coucher de soleil ou d'une nuit de pleine lune. Les bâtonnets et les cônes sont également effectifs. La notion de couleur est diminuée car les cônes ne sont pas assez stimulés, la vision périphérique est proportionnellement accrue par rapport à la vision centrale. C'est le domaine transitoire entre le domaine scotopique et le domaine photopique, il est difficile à exprimer car très variable d'un individu à l'autre.
- Domaine photopique : radiance supérieure à $\sim 10 \text{ cd.m}^{-2}$. Vision diurne ou lumière d'intérieur puissante. Utilisation principale des cônes situés dans la fovéa, d'où une forte définition des détails en vision centrale. Un phénomène de saturation se produit pour des intensités trop importantes pour l'oeil, c'est l'éblouissement.

1.3 Radiométrie et photométrie.

L'étude de la lumière, dans l'état actuel de nos connaissances, peut être effectuée de deux façons différentes, l'une en considérant la lumière comme une onde électromagnétique comme une autre, telles les ondes radios, les micro ondes, les ondes optiques, les rayonnements X, les rayonnements gamma : c'est la *radiométrie*. L'autre, la *photométrie*, étudie les radiations optiques perceptibles par l'oeil humain (fig. 1.5). La photométrie est équivalente à la radiométrie dans la gamme de fréquences visible par notre oeil, à la différence près que tout est pondéré par la réponse spectrale d'un oeil humain moyen, norme définie par la Commission Internationale de l'Éclairage (CIE) [dl83].

$$S_v(\lambda) = K_m \int_{380nm}^{680nm} S(\lambda)V(\lambda) d\lambda \quad \text{et} \quad S'_v(\lambda) = K'_m \int_{380nm}^{680nm} S(\lambda)V'(\lambda) d\lambda$$

Où K_m et K'_m valent respectivement 683 et 1700 lumen par Watt ($lm \cdot W^{-1}$).

La fonction de réponse spectrale dépend non seulement de la longueur d'onde mais aussi de l'intensité lumineuse perçue et des conditions d'expérimentations. La norme CIE définit cette fonction pour les domaines photopiques ($V(\lambda)$) et scotopiques ($V'(\lambda)$)(fig. 1.4). Chacune de ces fonctions est normalisée pour obtenir leurs maximums à $\lambda_m = 555 \text{ nm}$ pour $V(\lambda)$ et $\lambda'_m = 507 \text{ nm}$ pour $V'(\lambda)$ [Ash95, WS82]. Dans la réalité, il n'existe évidemment pas seulement deux courbes, mais une gerbe de courbes, qui sont dues à des phénomènes chimiques complexes au niveau des cônes et bâtonnets. Cette gerbe a très peu de chance d'être une interpolation linéaire des courbes $V(\lambda)$ et $V'(\lambda)$. Pour convertir une grandeur radiométrique (S_e) en grandeur photométrique (S_v), il faut intégrer sur le domaine photométrique en pondérant par la réponse spectrale de l'oeil.

* Les unités utilisées dans cette section sont les unités standards *candelas par mètre carré* (cd.m^{-2}) (cf. §1.4, page 10)

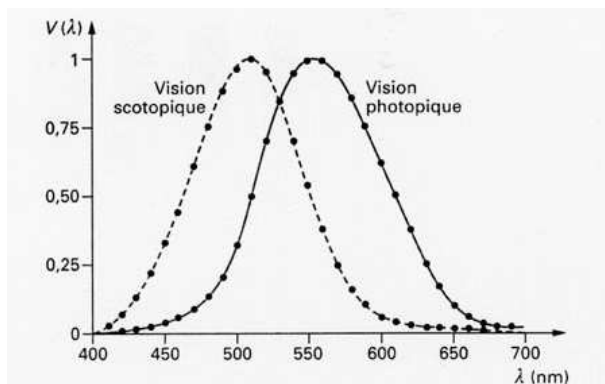


FIG. 1.4: Fonction de perception de l'intensité lumineuse de l'œil humain moyen $V(\lambda)$ et $V'(\lambda)$ respectivement pour les domaines photopiques et scotopiques.

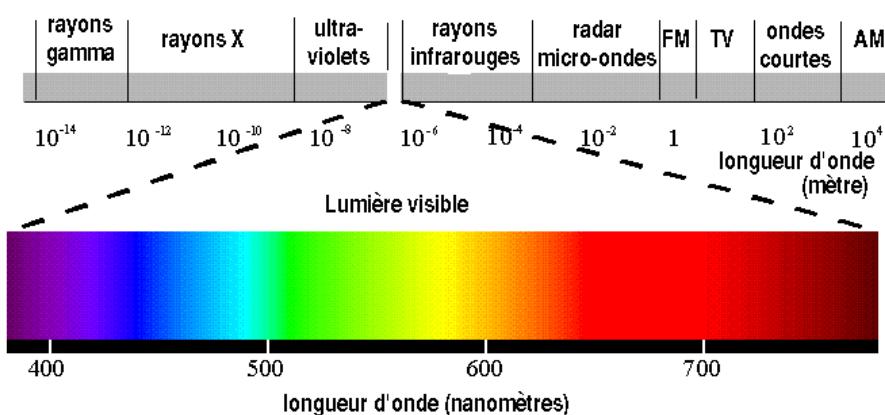


FIG. 1.5: Spectre des rayonnements électromagnétiques.

1.4 Unités de mesure

Les professionnels de l'éclairage utilisent différentes unités de mesure de l'énergie lumineuse qui se propage dans un environnement. Il existe souvent plusieurs unités différentes pour une même mesure car les standards ont été fixés tardivement. Certains termes, tel que "intensité", sont employés dans des contextes très différents, parfois de façon erronée. Dans ce chapitre, nous définissons les termes utilisés tout au long de cette thèse dans les domaines photométriques et radiométriques. Par convention, les grandeurs radiométriques sont indicées par un e , les grandeurs photométriques sont indicées par un v .

Énergie.

Ce terme est difficile à définir exactement et la majorité des définitions existantes ne donnent que des exemples. L'énergie est une grandeur qui représente la capacité d'un système à produire un travail à un instant donné. Si l'on considère la loi de conservation de l'énergie universelle, alors l'énergie est indéfinissable concrètement car il n'y a ni génération, ni destruction d'énergie. Certains parlent

de "souffle divin". L'énergie est contenue dans une onde électromagnétique sous forme électrique et magnétique. Par exemple l'énergie contenue dans une radiation de micro ondes est convertie en chaleur lorsqu'elle excite une molécule d'eau. Similairement un rayon lumineux transmet de l'énergie lorsqu'il vient exciter nos récepteurs visuels.

	Francophone	Anglophone	Symbole	Unité
Radiométrie	Energie	Radiant energy	Q_e	J (Joules)
Photométrie	Energie lumineuse	Luminous energy	Q_v	$lm \cdot s$ (lumen seconde) $\equiv talbot$

Puissance / flux.

La puissance d'un corps est définie par son énergie émise ou reçue par unité de temps dans toutes les directions. Par exemple une lampe de 100 *Watts* (régime constant) allumée pendant 1 minute émet $100 \cdot 60 = 6000 J$. On parle aussi de flux radiatif ou flux lumineux. Les utilisateurs de l'éclairage global parlent souvent indifféremment d'énergie ou de puissance car l'émission des surfaces est considérée en régime constant dans le temps, d'où l'équivalence de ces deux valeurs.

$$\Phi = \frac{dQ}{dt}, \quad Q \text{ est l'énergie de la source et } t \text{ l'unité de temps.} \quad (1.1)$$

	Francophone	Anglophone	Symbole	Unité
Radiométrie	Puissance radiative flux radiatif	Radiant power Radiant flux	Φ_e	W (Watt)
Photométrie	Puissance lumineuse	Luminous flux	Φ_v	lm (lumen)

Densité de flux / radiosité.

La densité de flux est la puissance émise ou reçue par unité de surface en provenance de toutes les directions d'un demi-espace (hémisphère). Le terme "densité de flux" n'est que très rarement utilisé mais regroupe l'émission et la réception d'un flux en un seul terme. On parle d'*irradiance* en cas de réception d'un flux radiatif et de *radiosité* en cas d'émission de ce flux en radiométrie. La photométrie utilise le terme *éclairage* indifféremment pour l'émission ou la réception. Il ne faut pas confondre l'algorithme de radiosité et la grandeur énergétique "radiosité", même s'ils sont regroupés tous les deux sous le terme de radiosité. Par exemple un plafonnier de surface $0.2 m^2$ d'une puissance de 100 *Watts* a une radiosité de $500 W \cdot m^{-2}$

$$M = \frac{d\Phi}{dA}, \quad A \text{ l'unité de surface.} \quad (1.2)$$

Réception :

	Francophone	Anglophone	Symbole	Unité
Radiométrie	Irradiance	Irradiance Radiant Power	M_e	$W \cdot m^{-2}$
Photométrie	Illuminance Eclairage	Illuminance	M_v	$lm \cdot m^{-2} \equiv lx$ (lux)

Émission :

	Francophone	Anglophone	Symbole	Unité
Radiométrie	Emittance Radiosité	Emittance Radiosity Radiant exitance Brilliance	B	$W \cdot m^{-2}$
Photométrie	Exitance Éclairement	Luminous exitance	M_v	$lm \cdot m^{-2} \equiv lx$ (lux)

Intensité.

L'intensité est utilisée dans des contextes très variés et fréquemment de façon erronée. Elle est définie par un flux par unité d'angle solide. Conceptuellement, elle représente la partie d'énergie (ou plus exactement de puissance) reçue par une aire infinitésimale d'une source ponctuelle dans une certaine direction. Une source monochromatique de fréquence $540 \cdot 10^{12} Hz$ qui a une intensité lumineuse (photométrie) de $1cd$ dans une direction donnée, a une intensité radiative (radiométrie) de $1/683 W \cdot sr^{-1}$ (cf. §1.3, page 9).

$$I = \frac{d\Phi}{d\omega \cdot \cos\theta} \quad (1.3)$$

où Φ est le flux, dA une surface élémentaire, $d\omega$ l'angle solide élémentaire et θ l'angle entre la normale à la surface et le flux.

	Francophone	Anglophone	Symbole	Unité
Radiométrie	Intensité (énergétique) Intensité radiative	Radiant Intensity	I_e	$W \cdot sr^{-1}$
Photométrie	Intensité lumineuse	Luminous Intensity	I_v	$lm \cdot sr^{-1}$ $\equiv cd$ (candela)

Radiance / luminance.

Le mot *radiance* n'existe pas dans la langue française. Nous l'introduisons ici car il est utilisé dans le domaine scientifique comme tel. L'équivalent français *luminance* est ambigu quand au domaine auquel il se réfère. La radiance est la densité de flux par unité d'angle solide. Cette grandeur ne distingue pas l'émission de la réception. La définition standard comprend l'émission, la réception et la transmission du flux au niveau de la surface.

$$L = \frac{d^2\Phi}{dA \cdot d\omega \cdot \cos\theta} = \frac{dI}{dA \cdot \cos\theta} \quad (1.4)$$

où Φ est le flux, dA la surface élémentaire, $d\omega$ l'angle solide élémentaire, I l'intensité et θ l'angle entre la normale à la surface et le flux.

	Francophone	Anglophone	Symbole	Unité
Radiométrie	Radiance Luminance énergétique	Radiance	L_e	$W \cdot m^{-2} \cdot sr^{-1}$
Photométrie	Luminance Luminance lumineuse	Luminous exitance Brightness	L_v	$lm \cdot m^{-2} \cdot sr^{-1}$ $\equiv nit$

La radiance est une quantité qui ne s'atténue pas avec la distance grâce à sa définition par unité d'angle solide. Elle est constante le long du chemin entre deux points x et y :

$$d^2\Phi = L_x \cdot dA_x \cdot d\omega_{xy} \cdot \cos\Theta_x$$

où $d\omega_{xy}$ est l'angle solide entre les deux surfaces élémentaires x et y .

$$d\omega_{xy} = \frac{dA_y \cdot \cos\Theta_y}{r^2}$$

En remplaçant dans l'équation précédente, nous obtenons :

$$\begin{aligned} d^2\Phi &= L_x \cdot dA_x \cdot \cos\Theta_x \frac{dA_y \cdot \cos\Theta_y}{r^2} \\ &= L_x \cdot dA_y \cdot \cos\Theta_y \frac{dA_x \cdot \cos\Theta_x}{r^2} \\ &= L_x \cdot dA_y \cdot \cos\Theta_y \cdot d\omega_{yx} \end{aligned}$$

Par définition nous avons :

$$d^2\Phi = L_y \cdot dA_y \cdot d\omega_{yx} \cdot \cos\Theta_y$$

En comparant les deux dernières équations, nous obtenons

$$L_x = L_y$$

La plupart des récepteurs lumineux, dont l'oeil et l'appareil photographique, sont sensibles à la radiance. Ce qui explique qu'un objet donne une impression visuelle identique quelle que soit sa distance.

1.5 Ordres de grandeurs réels.

Il est difficile pour le novice de se retrouver dans cet ensemble d'unités de mesures. Voici quelques exemples de valeurs des différentes grandeurs définies dans cette section.

Pleine lune	0,2 lx
Perception d'ambiance	de 2 à 50 lx
Perception grossière des détails	100 lx
Lecture	400 lx
Éclairage artificiel intense	de 10^3 à 10^4 lx
Soleil à 50° , ciel clair	10^5 lx

La puissance des sources lumineuses dépend de leur coefficient d'efficacité, une partie de l'énergie se dissipe en énergie calorifique et non lumineuse. Par exemple une lampe à incandescence de 100 W émet ~ 1200 lm alors qu'un tube fluorescent de 100 W émet ~ 5000 lm car son coefficient d'efficacité est supérieur à celui de la lampe à incandescence. le *coefficient d'efficacité k* (ou *efficacité lumineuse*) est défini par la quantité d'énergie lumineuse émise pour 1 Watt d'énergie (souvent électrique), il s'exprime donc en $lm \cdot W^{-1}$. L'efficacité lumineuse est aussi l'inverse de l'*efficacité énergétique*. Voici quelques exemples d'efficacité lumineuse de sources quotidiennes.

Lampe à incandescence basse température (traditionnelle)	10 à $15lm \cdot W^{-1}$
Lampe à incandescence halogène	15 à $30lm \cdot W^{-1}$
Lampe à induction	$65lm \cdot W^{-1}$
Lampe fluorescente (néons)	50 à $100lm \cdot W^{-1}$
Lampe à vapeur de mercure haute pression	$120lm \cdot W^{-1}$
Lampe à vapeur de sodium	$150lm \cdot W^{-1}$

2 Les algorithmes de rendu.

Les algorithmes d'éclairage global tendent à recréer l'équilibre énergétique lumineux d'un contexte en calculant l'ensemble des trajets des photons émis, réfléchis, absorbés, etc... Il serait possible d'étendre le domaine à d'autres types de radiations électromagnétiques, les radiations optiques ayant à peu près le même comportement que celles du domaine visible du spectre électromagnétique.

La génération d'images de synthèse est composée de deux parties ; la première est le calcul des échanges énergétiques et leur caractérisation, la seconde est la création de l'image proprement dite, c'est-à-dire le calcul de luminance de chaque pixel en fonction de la solution calculée précédemment. Ces deux phases ne sont généralement pas distinctes parce qu'il est possible de simplifier la résolution en combinant les deux étapes.

Cette section décrit, après une définition de ce qui est appelé *rendu*, les lois et les équations qui permettent de calculer l'éclairage d'une scène. Ensuite nous verrons les approximations qui ont été faites afin de pouvoir rendre ces équations calculables.

2.1 Rendu photoréaliste.

Le *rendu* est le domaine de la synthèse d'images qui calcule la luminance de chaque point de l'image à partir de la représentation géométrique de la scène, de la caméra et des différents éclairages. Il est possible d'obtenir différents types de rendus pour une même scène, le cinéma d'animation préférera un rendu style "bande dessinée" alors qu'un film cherchera à obtenir des images de synthèse les plus proches possible de la réalité (fig. 1.6).

Le *rendu photoréaliste* essaye de créer des images numériques indifférenciables d'un cliché photographique réel. Pour arriver à cette fin, il faut connaître, comprendre et savoir recréer les phénomènes physiques qui régissent la lumière. Ces phénomènes sont peu nombreux mais relativement complexes quand on les observe au niveau moléculaire. Les équations de Maxwell déterminent les "rebonds" des ondes électromagnétiques sur n'importe quelle surface. Les couches électroniques des atomes déterminent la couleur d'un matériau. Ces équations sont connues, mais il est impensable de les utiliser actuellement telles quelles car il faudrait alors une puissance de calcul hors d'imagination, par exemple, une lampe de 100 W émet 10^{27} photons à la seconde.

En regardant les lois régissant la lumière à un niveau supérieur d'abstraction, on simplifie les interactions tout en multipliant le nombre de cas possibles. Par exemple, la réflexion d'un rayon lumineux sur un miroir parfait est déterminée par la formule "angle incident = angle réfléchi". Certains phénomènes deviennent plus difficiles à caractériser tels que la phosphorescence, l'irisation, etc...



FIG. 1.6: Différents rendus pour une même géométrie.

2.2 Théorie.

L'équation de transport d'énergie [Gla95] basée sur les équations de Maxwell découpe le transport énergétique en cinq phénomènes différents : l'injection, l'émission, l'absorption, la déviation entrante et la déviation sortante [Arv93]. Ces équations sont trop complexes pour pouvoir les résoudre actuellement. De plus, certains phénomènes ne sont que très faiblement perceptibles, ce qui permet de les réduire. La solution théorique adoptée par le consortium de l'image de synthèse au problème d'éclairage global est l'équation du rendu (*rendering equation*) formalisé par Kajiya [Kaj86].

$$I(x, x') = g(x, x') \left[\varepsilon(x, x') + \int_S \rho(x, x', x'') I(x', x'') dx \right] \quad (1.5)$$

où :

{	$I(x, x')$ est la fonction d'échange énergétique entre le point x et x' ; $g(x, x')$ est un terme géométrique intégrant la visibilité entre x et x' ; $\varepsilon(x, x')$ est un terme énergétique relatif à l'émission propre entre x et x' ; $\rho(x, x', x'')$ est la fonction de réémission entre x'' et x en passant par le point x' .
---	---

Cette équation est une approximation du transport d'énergie réel car elle est une simplification des équations de Maxwell. Elle ne prend pas en compte la phase des ondes électromagnétiques, ni la participation du milieu entre x et x' .

2.3 Couleur.

Un rayon lumineux polychromatique est composé d'un ensemble continu ou discontinu de fréquences spectrales. Cette représentation est difficilement utilisable, il lui est préféré d'autres modèles de représentation du spectre lumineux :

- **Échantillonnage** : Discrétisation du spectre électromagnétique, la plupart du temps, uniquement la partie visible de 400 à 700 nm tous les 10 nm. Les calculs de couleur pour chaque rayon chromatique peuvent être groupés ou indépendants. Lorsqu'ils sont groupés, le rayon est polychromatique mais le gain de qualité est très faible par rapport à une base de couleur RVB (cf. ci dessous). Lorsqu'ils sont disjoints, chaque rayon est monochromatique, ce qui implique des calculs de transfert indépendants, soit une augmentation du temps de calcul proportionnelle au nombre de rayons utilisés pour représenter la couleur.
- **Base trichromique RVB** : Cette décomposition du spectre en une base tridimensionnelle (au sens mathématique) définit ses axes par les composantes Rouge, Verte et Bleu. Cette base est dite *additive* car la lumière blanche s'obtient par addition des trois composantes au noir (absence de lumière), ce qui correspond parfaitement à la simulation d'éclairage qui ajoute l'ensemble des irradiances en un point. Ce système est proche du système visuel humain. De plus, il est majoritairement utilisé pour la synthèse électronique des couleurs (écrans).
- **Base trichromique CMJ** : Les axes de cette base sont composés des couleurs Cyan, Magenta et Jaune, couleurs complémentaires du Rouge, Vert et Bleu. Cette base est *soustractive* car elle agit par filtration de la lumière blanche. La composition des trois composantes donne le noir. Elle est très mal adaptée à la synthèse d'image qui est additive, elle sert généralement dans l'impression qui se base sur un papier initial blanc.
- **Base psychovisuelle** : Cette base définit les couleurs par les trois axes : teinte, saturation et luminosité. La teinte est la position dans le spectre lumineux (fréquence de l'onde), ce qui équivaut à sa couleur. La saturation est la proportion de gris mélangé à la couleur pure. La luminosité détermine si la couleur est claire ou foncée. Cette base est facile d'utilisation pour définir des couleurs car elle est plus "naturelle" pour l'homme. Par contre cette base n'est pas

une transformation linéaire d'une synthèse additive ce qui la rend coûteuse pour l'utilisation en synthèse d'image.

La base la plus utilisée reste la base RVB car elle est facile d'utilisation et très proche du matériel. Par contre cela reste une approximation très grossière du spectre lumineux par un échantillonnage de trois valeurs. La meilleure méthode reste un échantillonnage plus fin, mais ceci au dépend du temps de calcul pour des résultats très légèrement plus proches de la réalité.

2.4 Fonction de réflectance bidirectionnelle.

La fonction de réflectance est le rapport de la quantité de lumière qui arrive sur une surface dans une direction donnée par rapport à la quantité de lumière réémise dans une autre direction. Une onde électromagnétique arrivant sur un matériau, traverse une couche superficielle (profondeur de peau), elle est déviée, réfractée, réfléchi et scindée dans cette zone. Elle ressort ensuite dans un ensemble d'autres directions.

Cette fonction liant la lumière entrante à la lumière sortante qui caractérise l'aspect que l'on voit de la surface du matériau : rugosité, couleur, spécularité, etc... L'ensemble des variables de cette fonction est :

1. position sur la surface ;
2. direction entrante et sortante ;
3. couleur entrante et sortante ;
4. polarisation (linéaire et circulaire) de l'onde électromagnétique entrante et sortante ;
5. profondeur de peau du matériau ;
6. décalage de position entre le point d'entrée et le point de sortie ;
7. temps de propagation de l'onde dans la surface.

Le nombre de paramètres est beaucoup trop élevé pour les prendre tous en compte en même temps. Les quatre derniers points cités interviennent dans des phénomènes physiques complexes et suffisamment rares pour être négligés. Restent donc la direction et la couleur sortantes en fonction de la direction et la couleur entrantes ainsi que la position sur la surface.

Si le matériau est considéré comme constant sur toute la surface de l'objet, la formulation devient indépendante de la position. Il reste alors une fonction de dimension 4 dont les variables sont les deux coordonnées sphériques de la direction entrante, et ses deux homologues sortantes. Cette fonction est l'approximation de base adoptée en imagerie de synthèse, elle est nommée *BRDF* (pour *Bidirectional Reflectance Distribution Fonction*). À cette fonction est souvent associée une texture de couleurs et/ou une texture de normales pour modifier certains paramètres séparables de la BRDF en fonction de la position spatiale sur la surface.

La composante colorimétrique est considérée séparable de la fonction de réflectance, effectivement le fait qu'une surface modifie de façon importante la teinte du faisceau lumineux incident reste un phénomène rare. La BRDF est donc définie pour chacune des composantes de la lumière.

BRDF diffuse.

La représentation la plus grossière de la BRDF est une réflectance constante dans toutes les directions (fig. 1.7), cela caractérise des surfaces mates comme le plâtre, le papier mat, etc... Ces surfaces sont dites *lambertiennes*. Le coût mémoire de cette représentation est une seule valeur par composante de couleur.

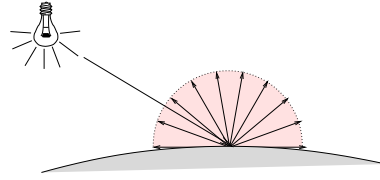


FIG. 1.7: BRDF diffuse caractérisant les surfaces lambertiennes.

BRDF à deux composantes.

Un second modèle d'approximation de cette BRDF la décompose en deux quantités distinctes, la première est la valeur diffuse de réémission dans toutes les directions, la seconde est une valeur spéculaire dépendante de l'angle d'incidence (fig. 1.8).

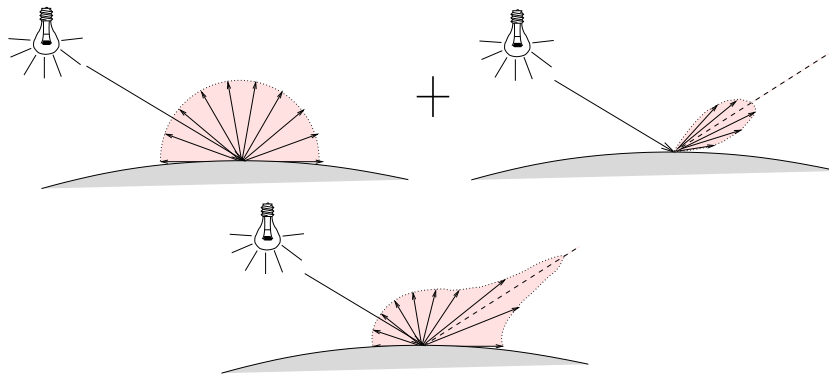


FIG. 1.8: BRDF à deux composantes, une diffuse et une spéculaire.

Cette décomposition a été très utilisée jusqu'à l'heure actuelle car elle permet de modéliser un grand nombre de matériaux à peu de frais. La classe de matériaux simulables est par contre très synthétique, et il faut recourir à des artifices pour éliminer ce phénomène.

BRDF quelconque.

La BRDF peut être très complexe quand la micro géométrie au niveau de la surface est compliquée. Un tissu de velours composé d'une multitude de petites fibres sera diffus en le regardant de face, alors que pour certains angles de vue, il aura une composante spéculaire non négligeable. Dans ce cas, l'approximation en 2 composantes est insuffisante, il faut utiliser d'autres méthodes plus précises. Certaines personnes ont calculé systématiquement les tables de la fonction bidirectionnelle par simulation [CMS87] ou encore les ont mesurées [Kri47, War92]. Cependant plusieurs régions dans ces fonctions sont lisses, et il devient possible de les coder en moins d'espace. Certains utilisent les harmoniques sphériques [WAT92, SAWG91], des notions de longueur sur une sphère géodésique [GMN94], ou des ondelettes sphériques [SS95].

Une nouvelle approche utilise une modélisation à six dimensions de la fonction de réflectance. Elle peut être, d'une certaine manière, considérée comme une généralisation directe du placage de texture et de la perturbation de normales. Cette modélisation est une sorte de texture de BRDF, les auteurs l'ont nommée *Bidirectional Texture Function (BTF)* [DvGNK99]. Le coût mémoire est d'autant plus exorbitant, et il reste de nombreux problèmes notamment dus à l'aliassage des textures, mais les

résultats sont très proches de la réalité et encourageants. Les BTF permettent d'intégrer l'introduction de défauts divers, tels que la porosité, les rayures et la corrosion, ces défauts ayant des conséquences importantes sur l'aspect visuel d'une surface.

3 Modèles d'éclairage local.

Les modèles d'éclairage local calculent la luminance d'un point d'une surface par éclairage direct par n sources ponctuelles. C'est une des bases du rendu (cf. §2.1, page 14) qui sera utilisé dans la majorité des algorithmes du domaine. Le terme français *modèle d'ombrage* est très peu utilisé : on lui substitue fréquemment le terme anglais *shading*.

3.1 Rendu lambertien.

Le *rendu lambertien* (*flat shading*) calcule une valeur de radiance en supposant la surface lambertienne (fig. 1.7). La valeur de la radiance ne dépend que de l'irradiance et de l'angle entre la source lumineuse et la normale à la surface (eq. 1.6).

$$I_\lambda = I_{l\lambda} C_d k_d (\vec{n} \cdot \vec{l}) \quad (1.6)$$

où I_l est l'intensité de la source lumineuse, C_d et k_d sont la couleur et le coefficient diffus, \vec{n} le vecteur normal à la surface et \vec{l} le vecteur d'éclairage de la source (fig. 1.9).

Cette technique de rendu lambertien est souvent utilisée en combinaison d'un algorithme de lissage afin de créer des transitions de couleur douces entre les différents polygones affichés. Les images obtenues semblent plus réalistes grâce à la continuité C^0 obtenue.

Nous allons voir deux types de lissage fréquemment utilisés pour leur bon rapport simplicité/gain visuel. Le premier est une interpolation des valeurs calculées aux sommets des polygones, le second est une interpolation des vecteurs normaux de chaque polygone.

Lissage de Gouraud.

Le *lissage de Gouraud* est une interpolation linéaire des valeurs de radiance (ou de couleur) entre les sommets de chaque polygone [Gou71]. Les valeurs aux sommets sont calculées comme la moyenne des valeurs des polygones auxquels le sommet appartient. L'adjonction du rendu lambertien et de l'interpolation de Gouraud est appelée le rendu de Gouraud (*gouraud shading*), cette dénomination est assez impropre car ce n'est pas un algorithme de rendu à part entière.

L'interpolation linéaire des valeurs de radiance limite l'utilisation de ce modèle à des surfaces lambertiennes car les variations induites par une BRDF dépendante du point de vue sont beaucoup plus fortes que ne le permet un maillage grossier.

Lissage de Phong.

Le *lissage de Phong* n'a jamais été décrit en temps que tel : la technique d'interpolation fait partie intégrante de son modèle de rendu [Pho75]. Elle est essentiellement liée aux capacités des ordinateurs à cette époque (années 75).

L'interpolation de Phong consiste à interpoler, non pas la couleur comme dans le lissage de Gouraud, mais les normales sur la surface du polygone. Les normales aux sommets sont calculées comme la moyenne des normales des polygones auxquels le point appartient. Ensuite, en chaque point du polygone, le calcul de la radiance est effectué en utilisant la normale interpolée.

Il y a souvent une confusion entre le modèle d'éclairage de Phong décrit dans la section suivante, et la technique d'interpolation des normales combinée avec un modèle de rendu. Quand le modèle utilisé est celui présenté par Phong, ceci est fréquemment appelé *Phong shading*.

3.2 Modèle de Phong.

Phong Bui-Tuong a décrit dès 1975 [Pho75] le modèle de la BRDF à deux composantes (cf. §2.4, page 17). Il utilise cette représentation pour en dériver une équation de rendu, qui, bien que simpliste, donne des résultats proches de la réalité dans la plupart des cas. Le calcul de radiance doit être exécuté en chaque point de l'image-résultat, et dépend du point de vue et de la normale à la surface.

Les deux parties de la décomposition de la BRDF donnent deux termes de radiance, l'un dit diffus, l'autre dit spéculaire. Cette lumière est ensuite "filtrée" par une fonction d'atténuation permettant de simuler médiocrement des effets de brouillard. Dans ce modèle, se rajoute aux deux termes, un troisième : le terme ambiant ; il est une approximation excessivement grossière de l'éclairage indirect, considéré comme constant en tout point.

$$I_\lambda = \underbrace{I_{a\lambda}k_aC_{d\lambda}}_{\text{ambiant}} + Att_\lambda \sum_i I_{l\lambda_i} \left[\underbrace{k_dC_{d\lambda}(\vec{n} \cdot \vec{l}_i)}_{\text{diffus}} + \underbrace{k_sC_{s\lambda}(\vec{r}_i \cdot \vec{v})^{n_s}}_{\text{spéculaire}} \right] \quad (1.7)$$

où I_l est l'intensité de la source lumineuse, $I_a k_a$ est l'intensité ambiante, C_d et k_d sont la couleur et le coefficient diffus. De même C_s et k_s sont la couleur spéculaire et le coefficient spéculaire, n_s est l'exposant spéculaire de réflexion. Att est une fonction d'atténuation, \vec{n} le vecteur normal à la surface, \vec{l}_i le vecteur d'éclairage de la source i , \vec{v} le vecteur d'observation et \vec{r}_i le vecteur d'éclairage idéalement réfléchi (fig. 1.9).

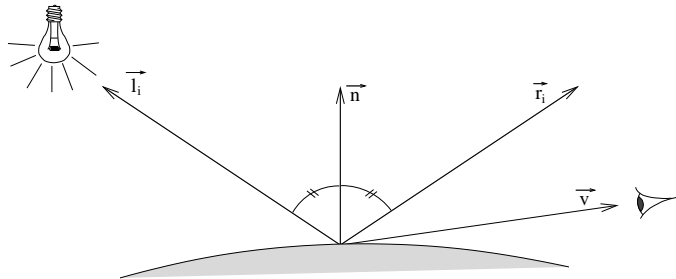


FIG. 1.9: Vecteurs utilisés dans le modèle d'éclairage lambertien du modèle d'éclairage de Phong.

4 Modèles d'éclairage global.

Les modèles d'éclairage global cherchent à estimer l'ensemble des chemins lumineux. Les modèles d'éclairage local permettent de calculer l'éclairage direct, la grosse difficulté étant d'estimer l'éclairage indirect en tout point. Deux grandes classes d'algorithmes ont été développées : la radio-sité et le suivi de rayons.

4.1 Chemins lumineux.

Heckbert a introduit une classification des rayons lumineux en utilisant le type d'interaction du rayon avec les surfaces qu'il rencontre de la source (L) à l'oeil (E). Ces interactions peuvent, selon lui, être dues à une réflexion diffuse (D) ou spéculaire (S) [Hec90]. Chaque rayon lumineux peut être caractérisé par un mot de cet alphabet contenu dans l'expression régulière $L(D|S)^*E$. Cette classification se base sur l'approximation induite par le modèle d'éclairage local de Phong qui sépare une composante spéculaire et une composante diffuse dans la BRDF, elle est donc valable uniquement dans ces cas-là.

4.2 Tracer de rayons.

Le tracer de rayon (*raytracing*) est la première technique d'éclairage global qui a été créée. On considère des milieux d'indices de réfraction homogènes dans lesquels la lumière se propage en ligne droite. Un rayon lumineux est donné par un point origine et un vecteur directeur. Lancer un rayon correspond à déterminer l'intensité et la couleur de la lumière transportée par ce rayon, en cherchant le chemin suivi par cette lumière et en effectuant des calculs en fonction des propriétés optiques des différents objets rencontrés.

Les chemins lumineux évalués sont du type LDS^*E . L'idée principale repose sur le parcours inverse de la lumière. On part de l'oeil et on "remonte" l'arbre des chemins lumineux arrivés à l'oeil en évaluant de façon récursive la radiance arrivant des directions réfléchies, réfractées et/ou transmises selon le cas, et ainsi de suite jusqu'à une surface diffuse. A chaque surface rencontrée, une évaluation de l'éclairage local produit par chaque source lumineuse est effectuée en prenant en compte la visibilité de celle-ci (rayon d'ombrage) (fig. 1.10).

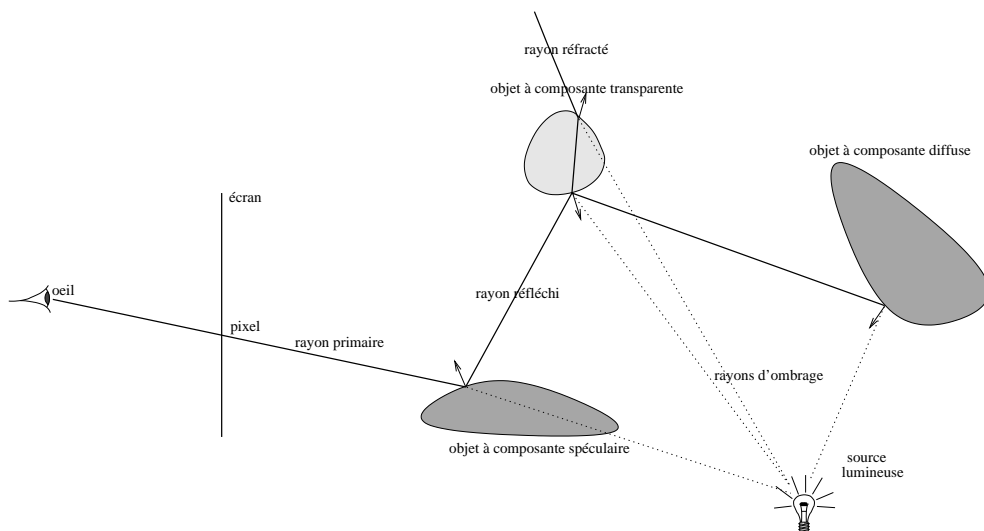


FIG. 1.10: Rayons lancés à partir de l'oeil pour calculer récursivement la radiance au pixel désiré.

Ainsi on ne calcule que la radiance des rayons lumineux passant par chaque pixel de l'image, ce qui limite les calculs à un nombre raisonnable. Cette technique très répandue repose sur des bases algébriques et géométriques relativement simples. Elle gère très bien les reflets de surfaces spéculaires, la transparence et la réfraction, l'éclairage direct spéculaire ou diffus de sources ponctuelles. Par contre, l'éclairage par des sources étendues, les reflets sur des surfaces non purement spéculaires

et les flous de bougé, ne peuvent pas être obtenus simplement. En utilisant une méthode de suréchantillonnage, il est possible de les simuler mais ceci à un coût en temps très élevé car le nombre de rayons augmente exponentiellement par rapport au nombre moyen d'intersections pour chaque rayon.

Le lancer de rayons est correct du point de vue géométrique pour le placement des rayons lumineux à l'intérieur d'une scène. Il est en revanche beaucoup moins précis du point de vue énergétique car les phénomènes que l'on désire étudier sont des transports d'énergies. Le lancer de rayons ne traduit qu'une partie de ces échanges lumineux.

Les transports d'énergies sont beaucoup plus compliqués que ceux que l'on peut modéliser par de simples calculs géométriques. Un avantage déterminant du lancer de rayons pour l'obtention d'images photoréalistes est que cet algorithme permet de rendre les réflexions et la transparence. De plus, l'ombrage est pris en compte intrinsèquement à chaque point d'évaluation de l'éclairage (pour des sources ponctuelles).

4.3 Tracer de chemins.

Le tracer de chemins (*Monte-Carlo path tracing*) est très proche du lancer de rayon. Cette technique utilise un algorithme de Monte-Carlo ou quasi Monte-Carlo pour résoudre l'équation de rendu (eq. 1.5). Un rayon est tiré à partir de l'oeil. À chaque interaction avec un objet, un seul rayon va être, au maximum, relancé dans une direction tirée aléatoirement ou pseudo-aléatoirement en fonction des propriétés du matériau. Ce rendu est très général, tous les différents types d'interaction peuvent être simulés [Hut93].

Par contre, comme dans une majorité des techniques basées sur Monte-Carlo, la solution est bruitée. Pour réduire le bruit, il faut augmenter le nombre d'échantillons, mais la convergence est lente. En utilisant un tirage purement aléatoire des rayons il faut simuler quatre fois plus de rayons, pour réduire la variance de moitié. Afin d'augmenter la vitesse de convergence, un échantillonnage d'importance (*importance sampling*) est souvent utilisé. Il permet d'obtenir plus probablement des rayons dont l'importance visuelle est élevée.

4.4 Tracer de photons.

Les techniques de tracer de photons sont apparues grâce à l'augmentation de la puissance de calcul. A l'inverse du tracer de rayon, le but est de trouver un chemin qui part des sources lumineuses et arrive à l'oeil [JC95]. Les premiers algorithmes de ce type furent appelés *lancer de rayon inverse* (*backward raytracing*) mais la confusion avec le principe du chemin inverse de la lumière a rendu ce terme désuet : on lui préfère les termes de tracer de photon (*photon maps*, *photon tracing*) ou *suivi de rayons* (*light tracing* ou *light path*).

La probabilité qu'un rayon lumineux partant d'une source passe exactement par l'oeil est quasiment nulle. Pour résoudre cela l'algorithme utilise deux passes. Dans un premier temps, les photons sont déposés sur les surfaces (diffuses). Comme dans la technique de tracer de chemins, une fonction probabiliste est utilisée pour créer le chemin du "photon" [DW94, Jen95, PP98]. La deuxième phase utilise les photons déposés afin de connaître la radiance au point désiré.

4.5 Lancer de rayon bidirectionnel.

Il est difficile de trouver les chemins lumineux partant exactement d'un point et arrivant exactement à un autre point, que ce soit de l'oeil vers une source ou d'une source vers l'oeil. La solution proposée avec le lancer de rayon bidirectionnel est de partir des deux points source et destination en même temps pour essayer de trouver les chemins lumineux qui peuvent les relier [CF87, LW93]. Les contributions des points d'intersection du rayon lumineux sont évaluées en chacun des points

d'intersection du rayon de l'œil en incluant la visibilité grâce à des rayons d'ombrage (fig. 1.11). Les chemins lumineux évalués sont du type $LS^*[D]S^*E$, donc limités à un seul échange diffus au maximum.

En se basant sur la même approche, Eric Veach ne connecte les rayons issus de la source et les rayons issus de l'œil que par un seul rayon d'ombrage à leurs extrémités ; les rayons d'ombrage sont calculés par un algorithme déterministe [VG94].

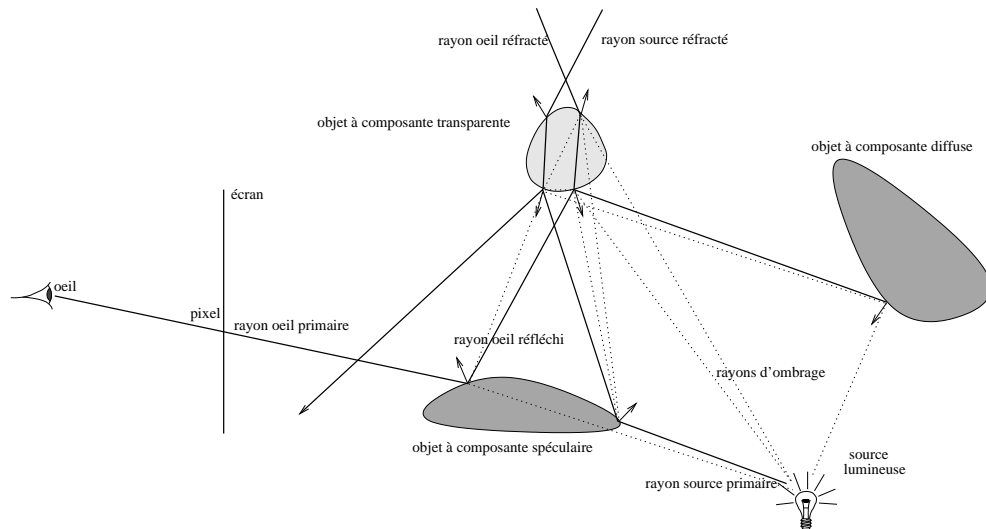


FIG. 1.11: Rayons lancés simultanément à partir de l'œil et la source pour calculer récursivement la radiance au pixel désiré.

4.6 Radiosité.

L'algorithme de radiosité est une méthode de résolution de l'équation de rendu (eq. 1.5) en utilisant des surfaces uniquement lambertiennes. Les chemins lumineux évalués sont du type LD^*E . L'énergie des sources est propagée dans la scène, c'est l'éclairage direct, puis chaque surface éclairée devient un nouvel émetteur et réémet une partie de son énergie, c'est l'éclairage indirect. Un des gros inconvénients de la radiosité est la nécessité d'avoir une représentation discrète de la scène, car tous les calculs de radiosité se font entre surfaces planes ou paramétriques. La radiosité est très puissante pour le calcul d'éclairage indirect par des surfaces purement diffuses (lambertiennes) et pour le calcul d'ombres douces dues à des émetteurs étendus (non ponctuels). Les solutions d'équilibre énergétique produites par les algorithmes de radiosité sont indépendantes du point de vue, ce qui les rend très utiles dans l'utilisation de mondes virtuels interactifs. Cette méthode est détaillée dans le chapitre 2 "Radiosité".

4.7 Méthodes hybrides.

Le lancer de rayons est efficace pour des scènes principalement spéculaires et la radiosité pour des scènes idéalement diffuses. Les tentatives de généralisation de ces deux méthodes aux domaines qu'ils ne savent pas traiter n'ont jamais donné de résultats très satisfaisants, les algorithmes devenant relativement lourds et moins précis.

Une autre classe de méthodes de calcul d'éclairage global utilise les deux algorithmes plus ou moins de façon séparée. Les premières approches cherchaient uniquement à réaliser une solution

utilisant l'aspect diffus des surfaces, puis dans un deuxième temps, ajouter à la solution, le calcul issu des surfaces spéculaires [WCG87, SP89, Gla95, SW99].

L'algorithme de radiosité

*Vois comme cette petite chandelle répand au loin sa lumière!
Ainsi rayonne une bonne action dans un monde malveillant.*

William Shakespeare

L'algorithme de radiosité, couramment appelé simplement *Radiosité* est un des deux algorithmes d'éclairage global utilisé actuellement. Cette simulation d'éclairage a connu un essor très rapide lors de ses premières applications à l'image de synthèse en 1984, majoritairement dû à sa capacité de traiter l'éclairage indirect. Les autres algorithmes de simulation d'éclairage estiment plus difficilement et à coûts plus élevés les réflexions multiples de la lumière dans une scène.

Dans un premier temps nous verrons l'adaptation de l'équation de rendu à la radiosité et la caractérisation de la notion de facteur de forme. Ensuite nous retracerons les différentes évolutions de l'algorithme de radiosité ainsi que les optimisations et variations autour de ce thème.

1 L'échange énergétique.

1.1 L'équation de luminance.

En ne considérant que des échanges lumineux entre surfaces dans un milieu homogène, l'équation de rendu définie par Kajiyà (eq. 1.5) peut s'écrire sous une autre forme, couramment appelée *équation de luminance*.

Soient :

$\left\{ \begin{array}{l} \mathcal{M} \text{ l'ensemble des points des surfaces de l'environnement} \\ \mathcal{S} \text{ l'espace des directions angulaires sur la sphère unité} \\ \mathcal{X} \text{ l'espace des fonctions à valeurs réelles définies sur } \mathcal{M} \times \mathcal{S} \end{array} \right.$

$$L(x, \omega) = L_e(x, \omega) + \int_{\Omega} k(x, \omega \rightarrow \omega') L(x', \omega') \cos \theta' d\omega' \quad (2.1)$$

avec $\left\{ \begin{array}{l} L, L_e \in \mathcal{X} \text{ les fonctions de luminance et d'auto-émission.} \\ \Omega \text{ la sphère des directions entrantes ou sortantes.} \\ k \text{ la fonction de réflectance bidirectionnelle.} \\ \theta' \text{ l'angle du rayon incident contenu dans l'angle solide } d\omega' \\ x \text{ le point de calcul de la radiosité.} \\ x' \text{ le point de la surface la plus proche déterminé par } \omega' \text{ et } x \\ \omega, \omega' \in \Omega \text{ respectivement les directions sortantes et entrantes.} \end{array} \right.$

La luminance en un point x et dans une direction ω est égale à la luminance propre d'émission ($L_e(x, \omega)$) à laquelle on rajoute la contribution lumineuse de chaque direction incidente ω' ($L(x', \omega')$) pondérée par la fonction de réflectance bidirectionnelle (cf. §2.4, page 16) et le cosinus de l'angle entrant.

Cette équation intégrale représente un système d'équations de dimension infinie sans solution analytique connue. Il faut donc utiliser des méthodes de résolution d'intégrales complexes approchant le plus possible la solution théorique.

La radiosité ne considère que des surfaces lambertiennes, ce qui permet d'en déduire les simplifications suivantes :

$$\left\{ \begin{array}{l} L(x, \omega) = \frac{B(x)}{\pi} \\ L_e(x, \omega) = \frac{E(x)}{\pi} \\ k(x, \omega' \rightarrow \omega) = \frac{\rho(x)}{\pi} \end{array} \right.$$

avec $B(x)$ la radiosité au point x , $E(x)$ la luminance propre au point x , et $\rho(x)$ la réflectance en x .

Grâce à l'indépendance de la fonction de réflectance par rapport à la direction entrante, il est possible de sortir $\rho(x)$ de l'intégrale, et en utilisant la relation entre l'angle solide élémentaire et l'aire projetée du récepteur (fig. 2.1).

$$d\omega' = \frac{\cos \theta' dA_j}{\|x - x'\|^2}$$

L'équation de luminance (eq. 2.1) s'écrit alors :

$$B(x) = E(x) + \rho(x) \int_{A_j} B(x') \frac{\cos \theta \cos \theta' \mathcal{V}(x, x')}{\pi \|x - x'\|^2} dA_j \quad (2.2)$$

où $\mathcal{V}(x, x')$ est la fonction de visibilité entre x et x' . Elle vaut 1 si les deux points sont visibles, et 0 s'ils sont occultés.

Cette équation, appelée *équation de radiosité*, traduit l'équilibre des échanges lumineux dans une scène, c'est une restriction de l'équation de rendu (eq. 1.5). Contrairement aux algorithmes basés sur des méthodes de Monte-Carlo, les divers algorithmes de radiosité utilisent une discrétisation des surfaces pour estimer les échanges radiatifs entre elles en utilisant une méthode de collocation [AC97] ou d'éléments finis.

Initialement et principalement pour des raisons puissance de calcul, la radiosité a été décrite avec des fonctions de radiosité constantes [GTGB84] ce qui permet de sortir le terme $B(x)$ de l'intégrale. Il en résulte un terme purement géométrique à l'intérieur de l'intégrale qui joue un grand rôle dans l'estimation de la valeur de la radiosité : *le facteur de forme*.

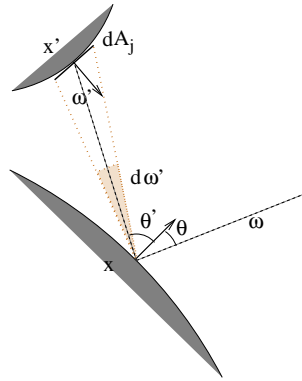


FIG. 2.1: Géométrie du transfert énergétique.

1.2 Facteurs de forme

Il existe dans la littérature de transfert thermique une grande variété de facteurs de forme dans des conditions géométriques particulières [How82, Lov68, HS67, Lov68]. Par exemple, le facteur de forme d'un cylindre vers un autre type de surface est intéressant pour calculer précisément l'échange énergétique entre un tuyau de chauffage et d'autres surfaces. L'algorithme de radiosité, en l'état actuel des choses, nécessite la discrétisation des objets en éléments de surface planaires rendant obsolète un certain nombre de formules, seul un petit ensemble des formules les plus usitées sont présentées.

On note F_{A_i, A_j} le facteur de forme entre une surface A_i et une autre A_j . Voici deux propriétés intéressantes des facteurs de forme :

- **Propriété 1** : Dû à la symétrie de l'intégrale $\|A_i\|F_{A_i, A_j} = \|A_j\|F_{A_j, A_i}$
- **Propriété 2** : Dans un environnement clos $\sum_j F_{A_i, A_j} = 1$

Dans les formules de facteur de forme suivantes, les surfaces A_i et A_j sont situées respectivement aux points p_i et p_j et les normales en ces points sont respectivement \vec{n}_i et \vec{n}_j . La distance p_i, p_j est notée r . Les deux angles entre le segment $\overline{p_i p_j}$ et les deux normales \vec{n}_i et \vec{n}_j sont définis par :

$$\theta_i = (\vec{n}_i, \widehat{p_i p_j}) \quad \text{et} \quad \theta_j = (\vec{n}_j, \widehat{p_j p_i})$$

On note aussi $\mathcal{V}(p_i, p_j)$ la fonction de visibilité entre le point p_i et p_j , elle vaut 1 quand les points sont visibles et 0 quand il y a un obstacle opaque entre les deux. Elle est fréquemment sortie de l'intégrale du facteur de forme pour une question de simplification de calculs, cette approximation est relativement faible pour des émetteurs uniformes.

Point - point.

Facteur de forme d'une surface élémentaire dA_i à une surface élémentaire dA_j .

$$F_{dA_i, dA_j} = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} \mathcal{V}(p_i, p_j) dA_j$$

Point - surface.

Facteur de forme d'une surface élémentaire dA_i à une surface A_j

$$F_{dA_i, A_j} = \int_{p_j \in A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} \mathcal{V}(p_i, p_j) dA_j$$

Point - polygone.

Facteur de forme d'une surface élémentaire dA_i à un polygone A_j dans le cas de visibilité totale. A_j est composé de N sommets $p_{j_1} \dots p_{j_N}$ (fig. 2.2)

$$F_{dA_i, A_j} = \frac{1}{2\pi} \sum_{n=1}^N \beta_k \vec{n}_i \frac{\overrightarrow{p_i p_{j_n}} \wedge \overrightarrow{p_i p_{j_{n+1}}}}{\|\overrightarrow{p_i p_{j_n}} \wedge \overrightarrow{p_i p_{j_{n+1}}}\|}$$

où β_k est l'angle entre le vecteur $\overrightarrow{p_i p_{j_n}}$ et le suivant $\overrightarrow{p_i p_{j_{n+1}}}$ exprimé en radians.

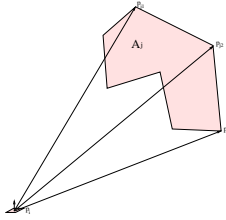


FIG. 2.2: Facteur de forme entre une surface élémentaire et un polygone.

Point - disque.

Facteur de forme d'une surface élémentaire dA_i à un disque A_j parallèle au plan de la surface éloigné d'une distance h (fig. 2.3a) :

$$F_{dA_i, A_j} = \frac{\|A_j\| \cos\theta_i \cos\theta_j}{\|A_j\| + \pi r^2}$$

Si le centre est aligné avec l'axe de la surface élémentaire A_i défini par (p_i, \vec{n}_i) (fig. 2.3b) la formule devient :

$$F_{dA_i, A_j} = \frac{r^2}{h^2 + r^2} = \frac{\|A_j\|}{\|A_j\| + \pi h^2}$$

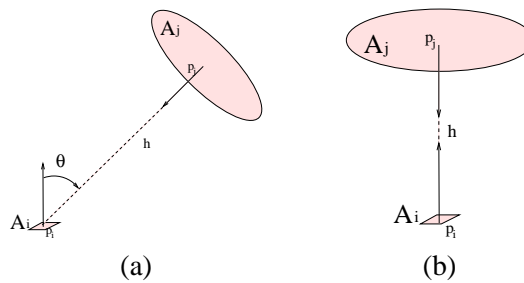


FIG. 2.3: Facteur de forme entre une surface élémentaire et un disque.

Point - sphère.

Facteur de forme d'une surface élémentaire dA_i à une sphère A_j éloignée d'une distance h (fig. 2.4a) :

$$F_{dA_i, A_j} = \frac{1}{2} \left[1 - \sqrt{1 - \frac{r^2}{h^2}} \right]$$

Si le centre de la sphère appartient à la droite (p_i, \vec{n}_i) (fig. 2.4b) :

$$F_{dA_i, A_j} = \frac{r^2}{h^2}$$

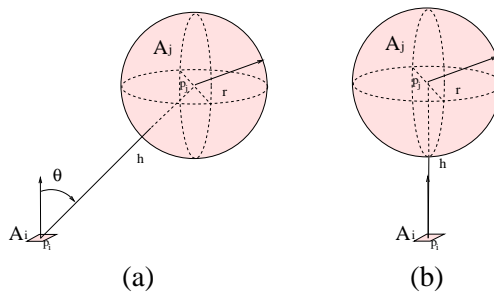


FIG. 2.4: Facteur de forme entre une surface élémentaire et une sphère.

Point - environnement.

Le calcul de forme est la tâche la plus coûteuse en temps de calcul de l'algorithme de radiosité car elle intègre la visibilité (cf. chapitre 5, page 97). La première technique utilisée pour limiter le nombre d'évaluations pré-calculé en un point p la totalité des facteurs de formes de ce point vers l'ensemble de l'environnement visible. Cette technique connue sous le nom d'hémicube [CG85] consiste à projeter l'environnement sur un demi cube discret centré en p , soit cinq plans de projection (fig. 2.5a).

Les projections sont relativement rapides grâce à l'utilisation de l'algorithme de tampon de profondeur [Cat74, Cat75] qui, de plus, est dorénavant câblé dans les cartes graphiques. Un des gros avantages est la prise en compte intrinsèque de la visibilité grâce à l'élimination des faces cachées du tampon de profondeur. Par contre, ce calcul souffre de la forte dépendance à l'aliassage des techniques discrètes.

La valeur du facteur de forme d'une surface est la somme des facteurs de forme des pixels projetés sur l'hémicube. Ces facteurs de formes sont très simples dus à la géométrie cubique orthonormée par rapport au point et à la normale à la surface émettrice (fig. 2.5b). Les facteurs de forme valent

$$\begin{cases} \Delta F = \frac{\Delta A}{\pi(1+y^2+z^2)^2} & \text{pour les pixels latéraux} \\ \Delta F = \frac{\Delta A}{\pi(1+x^2+y^2)^2} & \text{pour les pixels frontaux} \end{cases}$$

Il existe plusieurs variations sur ce thème, en utilisant un tétraèdre à la place d'un cube [Ash95], en utilisant un seul plan plus proche si l'on admet que la contribution des plans latéraux est très faible [SP89]. Enfin, en n'utilisant toujours qu'un seul plan, mais avec deux résolutions de façon à gagner de la précision pour les échanges énergétiques frontaux [RGG90].
(reproduction avec l'accord de Cyril Soler)

Surface - surface.

Facteur de forme d'une surface A_i à une surface A_j .

$$F_{A_i, A_j} = \frac{1}{\|A_i\|} \int_{p_i \in A_i} \int_{p_j \in A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} \mathcal{V}(p_i, p_j) dA_j dA_i$$

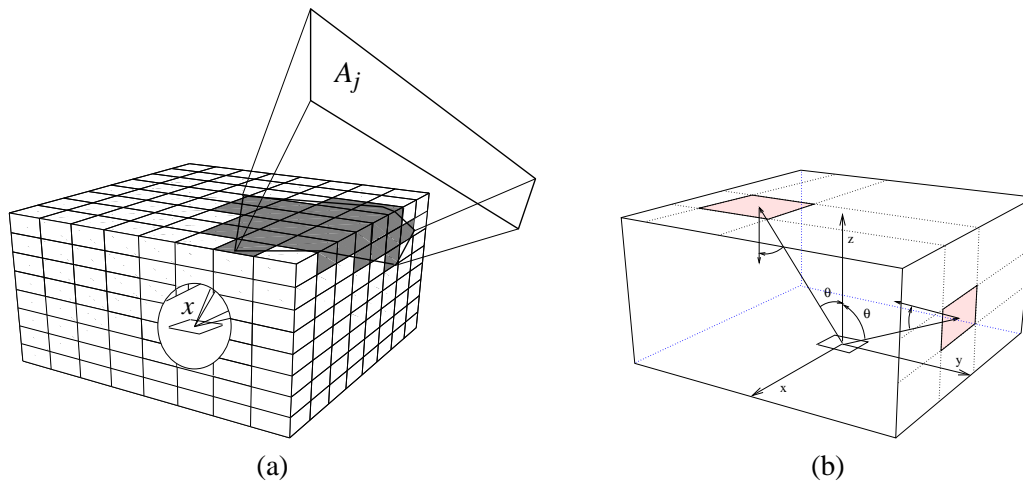


FIG. 2.5: L'hémicube : calcul discret du facteur de forme d'un point vers l'environnement visible de ce point.

Monte-Carlo.

Le calcul intégral du facteur de forme peut s'effectuer par des techniques d'intégration numériques de fonctions complexes telles que les techniques de Monte-Carlo. Les rayons sont lancés stochastiquement à partir du point ou de la surface où l'on désire calculer le facteur de forme vers le reste de l'environnement [MBG86, Mal88]. Philippe Bekaert décrit différentes méthodes basées sur l'échantillonnage des surfaces et des directions que ce soit de façon locale à la surface traitée ou globalement par rapport à la scène [Bek99].

La fonction de facteur de forme n'étant pas uniforme sur l'espace des directions il est possible d'utiliser une fonction d'importance pour améliorer la précision du calcul [SP95]. Cette approche peut-être appliquée pour des surfaces planes ou courbes. Il est aussi possible de prendre en compte des effets de transparence ou de surfaces non diffuses. La distribution aléatoire des échantillons assure la minimisation des effets d'aliasage et les techniques d'optimisation de lancer de rayon peuvent être appliquées [AK89, FvFH90, Gla95]. Le seul désavantage est le nombre élevé de rayons qui doivent être lancés pour arriver à la précision fournie par l'hémicube et ses dérivés.

2 Les algorithmes de radiosit .

L'algorithme de radiosit  doit estimer la fonction int grale d crite par l' quation de luminance (eq. 2.1). Il existe plusieurs techniques d'estimation d'int grales, nous verrons comment elles ont  t  appliqu es sp cifiquement pour le probl me qui nous int resse. La r solution directe, diff rentes relaxations et une estimation par des techniques de Monte-Carlo seront pr sent es. Nous nous attarderons  galement sur la formulation de Galerkin qui a l'avantage de regrouper l'ensemble des r solutions discr tes.

Ensuite nous verrons les diff rentes recherches effectu es autour de ces algorithmes et de leurs optimisations, en nous attardant sur les outils sp cifiques aux solutions hi rarchiques.

2.1 Radiosité de Galerkin

La formulation de *Galerkin* de l'équation de radiosité a été introduite par Heckbert [Hec91] et Zatz [Zat93], une dizaine d'année après l'apparition des premiers algorithmes. C'est une généralisation de la formulation initiale discrète considérant les fonctions de radiosité constantes sur chaque surface. La radiosité de Galerkin est définie sur un ensemble de surfaces constituant la scène, dont la fonction $\tilde{B}(p)$ est une approximation de la fonction de radiosité $B(p)$ sur ces éléments. Cette approximation est une projection de la fonction de radiosité réelle sur une base de fonctions quelconques $\{\psi_i\}$. La fonction de radiosité d'une surface est alors représentée par une combinaison linéaire des fonctions de cette base : $\tilde{B}(p) = \sum_i \alpha_i \psi_i(p) \simeq B(p)$.

La base de fonctions définit l'espace fonctionnel dans lequel on cherche à représenter B . C'est elle qui va conditionner la précision de représentation maximale de cette fonction, mais aussi les temps de calculs et la place mémoire utilisée pour stocker les coefficients α_i . Il est théoriquement possible d'utiliser une base de fonctions différente pour chaque surface, en pratique, une seule base est utilisée pour la totalité de la solution de radiosité.

Dans ce formalisme, les premiers algorithmes de radiosité utilisent une base de fonctions composée d'une unique fonction porte ψ_0 . Soit, effectivement, une valeur de radiosité constante sur tout l'élément.

$$\psi_0(p) = \begin{cases} 0 & \text{si } p \text{ appartient à la surface} \\ 1 & \text{si } p \text{ n'appartient pas à la surface} \end{cases}$$

2.2 Résolution matricielle

La résolution matricielle considère une scène composée de surfaces dont la représentation de la fonction de radiosité sur chacune est constante. Si l'on note M la matrice $n \times n$, de terme général $m_{ij} = \rho_i F_{ij}$ avec n le nombre de surfaces de la scène, le système d'équations de radiosité peut s'écrire sous forme matricielle

$$B = E + MB \iff B = E(I - M)^{-1} \quad (2.3)$$

où B est le vecteur des radiosités de chaque élément de surface, et E le vecteur de leurs luminances propres. Ce système admet une solution unique car le rayon spectral de la matrice M est strictement inférieur à 1. Grâce à la propriété 2 (cf. §1.2, page 27) la matrice $I - M$ est inversible, d'où l'équivalence de l'équation (eq. 2.3) [Atk89].

Il est donc théoriquement possible de trouver la solution d'équilibre énergétique exacte de cette équation, mais la complexité informatique des méthodes de résolution directes sont en $O(n^3)$ ce qui ne les rend applicables que pour des scènes d'une très faible complexité. L'exactitude de la solution n'étant pas nécessaire on utilise des méthodes itératives approximatives de résolution moins coûteuses. C'est la *radiosité progressive*.

2.3 Radiosité progressive

Tout comme dans la méthode de résolution matricielle, les méthodes de radiosité progressive utilisent une discrétisation de la scène en surfaces et considèrent une radiosité constante sur celles-ci. Parce que leur interprétation physique est simple, la radiosité progressive a été étendue à des fonctions de radiosité complexes, même si les méthodes de résolution matricielles ne sont plus applicables dans ce cas.

La radiosité des surfaces est fixée à zéro, puis l'algorithme calcule itérativement des nouvelles valeurs de fonction de radiosité pour un ensemble de surfaces, ceci jusqu'à convergence de la suite. Les deux méthodes classiques de résolution itérative d'inversion de matrice de *Jacobi* et *Gauss-Seidel*

ont une complexité en $O(n^2)$, ce qui les a rendu tout indiquées à la résolution de l'équation de radiosité. En pratique, la méthode de Gauss-Seidel est préférée à celle de Jacobi parce qu'elle répartit plus rapidement l'énergie dans la scène au cours des premières itérations. L'interprétation de la résolution de ce système est simple, à chaque itération on effectue successivement $B_i = E_i + \rho_i \sum_j F_{ij} B_j$, c'est-à-dire que l'on ajoute à chaque surface A_i la contribution énergétique de toutes les autres surfaces A_j . Cela représente une collecte des énergies incidentes (fig. 2.6b). Cette méthode, appelée radiosité *par collecte* (*gathering*), nécessite de calculer l'échange énergétique de toutes les surfaces pour obtenir une première image de la fonction de radiosité.

Une seconde méthode de résolution itérative peut-être utilisée, c'est la méthode de *Southwell*. Elle admet une interprétation physique duale, c'est-à-dire qu'à chaque itération, la surface A_i émet son énergie vers la totalité des autres surfaces (fig. 2.6a) $\forall j, \beta_j = \beta_j + \rho_i F_{ij} r_i$, où β_i est l'énergie accumulée sur la surface i et r_i le résidu de la relaxation de Southwell équivalent à l'énergie non encore émise. Cette méthode, appelée *par propagation* (*shooting*), permet d'avoir une estimation de l'éclairage primaire très rapidement, car les surfaces ayant une luminance propre émettent leur énergie dans toute la scène en une seule itération.

Une fois cette énergie répartie dans toute la scène, la convergence est très lente car quasiment toutes les surfaces de la scène deviennent émettrices. Pour augmenter la vitesse de convergence, il est possible de modifier la solution initiale [], ou d'émettre plus d'énergie durant les échanges énergétiques [FP92, Fed95]. Il est possible de voir l'évolution de la solution étape par étape, mais les premières images sont très sombres parce que la totalité de l'énergie n'a pas encore été émise. En ajoutant uniformément l'énergie non encore propagée (soit la somme des résidus $\sum_{i=1}^N r_i$) sur la scène, l'estimation de la solution est moins grossière [CCWG88].

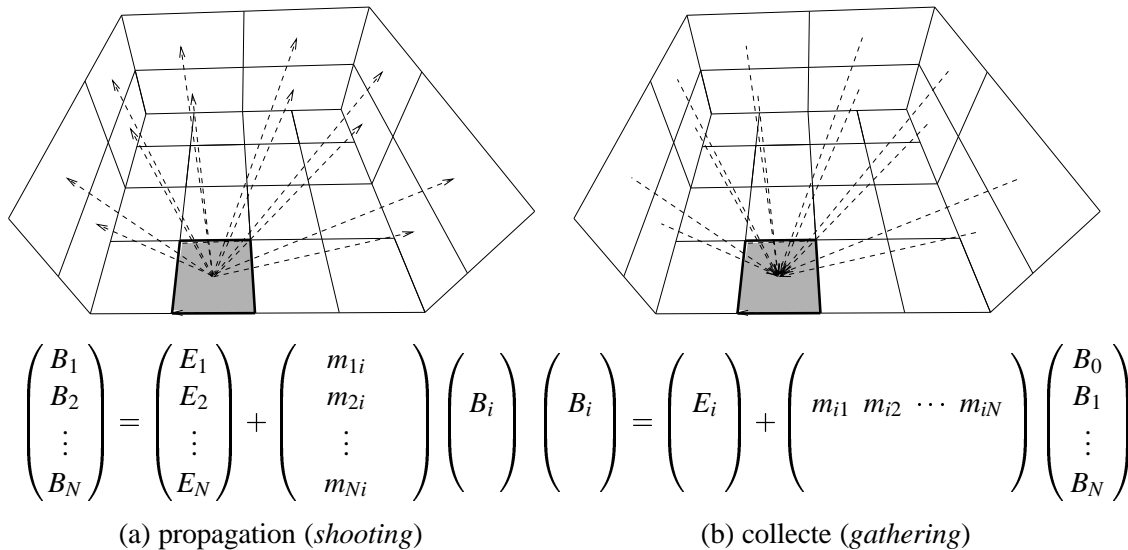


FIG. 2.6: Résolution de l'équation de radiosité par les méthodes de Southwell (a) et Gauss-Seidel (b).

Si les fonctions de radiosité ne sont pas constantes sur les surfaces, on ne peut pas explicitement parler de relaxation de Southwell ou de Gauss-Seidel, mais l'analogie de propagation (*shooting*) et de collecte (*gathering*) est conservée. L'algorithme de radiosité progressive considère les échanges énergétiques de chaque surface vers toutes les autres surfaces, ce qui lui confère sa complexité en $O(n^2)$ avec n le nombre de surfaces de la scène. Si l'on désire une représentation fine de la fonction

de radiosité, il est nécessaire d'avoir une discrétisation élevée de la scène, donc un nombre de surfaces important, donc des temps de calcul trop élevés pour des scènes complexes.

2.4 Radiosité Hiérarchique

Dans une scène très vaste, une grande partie des échanges énergétiques sont très faibles voire négligeables. Imaginez l'interaction entre deux petits morceaux de murs opposés dans un vaste hall de gare, peut-être serait-il suffisant de calculer l'interaction entre les deux murs entiers. D'un autre côté le calcul de l'interaction entre une lampe de bureau et la totalité du plateau de la table sur laquelle elle est posée, peut-être trop imprécise car le bout du plateau peut, par exemple, ne pas être éclairé.

L'idée de la radiosité hiérarchique est d'avoir une représentation de la fonction de radiosité à tous les niveaux hiérarchiques, du plus grossier au plus fin, là où c'est nécessaire, afin de pouvoir choisir la représentation la plus adéquate à utiliser dans le calcul de chaque interaction (fig. 2.7).

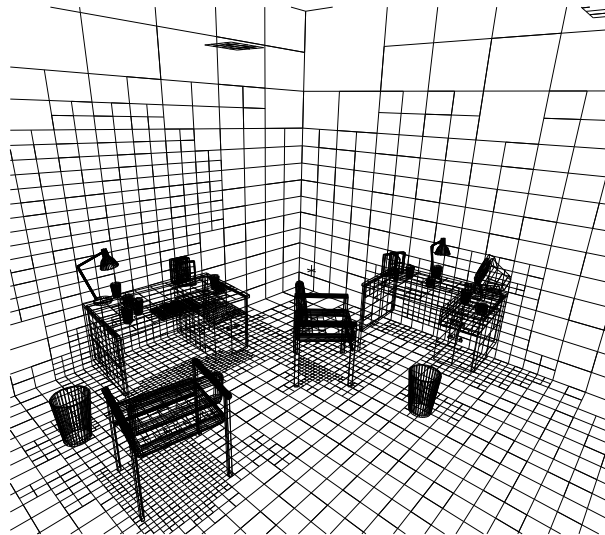


FIG. 2.7: Exemple de subdivision localisée dans les zones d'intérêts.

Soit H le nombre de niveaux hiérarchique et Ψ l'ensemble des bases de fonctions $\{\psi_i^{1..H}\}$:

$$\Psi = \{ \{\psi_1^1, \dots, \psi_{n_1}^1\}, \{\psi_1^2, \dots, \psi_{n_2}^2\}, \dots, \{\psi_1^H, \dots, \psi_{n_H}^H\} \}$$

où n_h est le nombre de surfaces nécessaires à la représentation de la fonction de radiosité au niveau hiérarchique h ($n_h = 4^h$ pour un quadtree).

La fonction de radiosité a H niveaux de représentation $\{\tilde{B}^1 \dots \tilde{B}^H\}$ du plus grossier au plus fin :

$$\forall h \in [1..H], \tilde{B}^h(p) \simeq \sum_{i=1}^h \alpha_i^h \psi_i^h(p)$$

Plus on descend dans la hiérarchie, plus l'estimation de la fonction est précise : $\|\tilde{B}^{h+1} - B\| \leq \|\tilde{B}^h - B\|$. Par contre il est rare que la fonction de radiosité puisse être exactement représentée dans la base de fonctions quel que soit le niveau hiérarchique, habituellement $\|\tilde{B}^\infty - B\| > 0$.

Même si le terme radiosité hiérarchique comprend toutes les bases de fonctions possibles (cf. §2.1, page 31), il est fréquent que ce terme soit utilisé pour la base de la fonction constante (ordre 0), les

fonctions d'ordre élevé sont connues sous le nom de radiosité par ondelettes (cf. §2.6, page 36). La figure (fig. 2.8) représente une fonction de radiosité (en 1 dimension) à différents niveaux hiérarchiques sur une base de fonctions constantes.

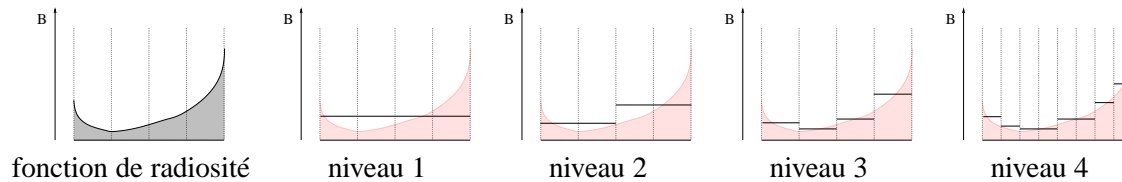


FIG. 2.8: Représentation d'une fonction de radiosité à différents niveaux hiérarchiques.

La radiosité hiérarchique à un coût en temps réduit à $O(n \log n + s^2)$ avec s le nombre de surfaces initiales et n le nombre de surfaces après subdivision.

Liens.

Les liens dans la radiosité hiérarchique symbolisent les échanges énergétiques et caractérisent les acteurs (émetteurs et récepteurs) de cet échange. Un lien établi entre deux éléments hiérarchiques supporte le facteur de forme et la visibilité les reliant. En conservant la totalité des liens d'une scène et avec la valeur d'émission de chaque surface, il est possible de reconstruire la totalité de la solution de la fonction de radiosité. Si la radiosité d'une surface change, il est uniquement nécessaire de vérifier que l'énergie portée par le lien n'est pas devenue trop imprécise, dans ce cas il faut supprimer ce lien pour en établir d'autres à des niveaux inférieurs plus précis.

Ces liens prennent une grande partie de la mémoire utilisée par la solution de radiosité, mais leur stockage n'est pas nécessaire. Si l'on utilise un algorithme basé sur la propagation d'énergie (cf. §2.3, page 31), alors, au prix de calculs parfois redondants, il est possible de supprimer le stockage des liens [SSSS98] et ainsi pouvoir traiter (plus lentement) des scènes beaucoup plus imposantes en nombre de polygones.

Push/pull.

A chaque itération de l'algorithme de radiosité, l'échange énergétique des liens est évalué à leur niveau. Pour conserver la cohérence énergétique des différents niveaux hiérarchiques, il est nécessaire de propager ces énergies dans l'ensemble de la hiérarchie. Ceci est effectué par un algorithme de *push/pull* qui "descend" l'énergie apportée par les liens aux feuilles de la hiérarchie, puis la "remonte" en la moyennant sur les parents jusqu'à la racine (fig. 2.9). En procédant ainsi, chaque niveau hiérarchique i a une estimation de la fonction de radiosité plus précise que le niveau $i - 1$ tout en ayant le maximum d'information possible à son niveau.

Depuis l'introduction de l'utilisation de textures dans la radiosité hiérarchique, les valeurs descendues durant le *push* sont des irradiances, alors que les valeurs remontées lors du *pull* sont des énergies. Ceci pour prendre en compte les différences de couleur sur un même support [GSH94].

Raffinement.

Une des grosses difficultés des approches hiérarchiques réside dans le choix des niveaux hiérarchiques d'établissement des échanges énergétiques. Ce choix est effectué par un *oracle de raffinement*. L'oracle est une heuristique qui définit la qualité, la précision, et les temps de calcul de la solution. Le

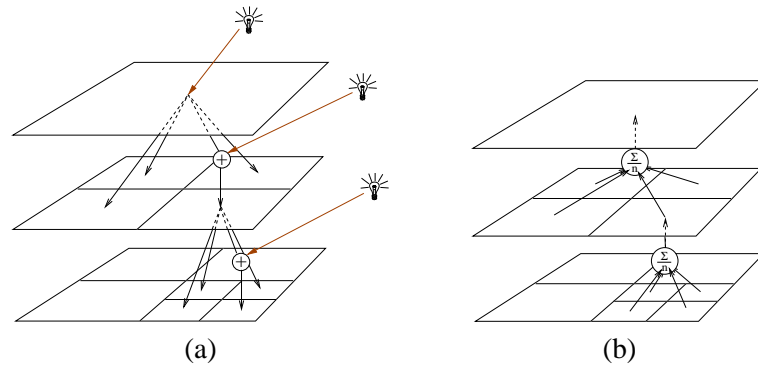


FIG. 2.9: Push-Pull : Conservation de la cohérence énergétique dans la hiérarchie. (a) Diffusion de l'irradiance du haut de la hiérarchie jusqu'aux feuilles. (b) Moyenne des valeurs d'énergie lumineuse des feuilles jusqu'au sommet.

critère d'estimation de la précision est théoriquement la différence entre la radiosité estimée et la radiosité réelle $\|\tilde{B} - B\|$, or évidemment, nous ne connaissons pas la valeur de la radiosité réelle puisque nous cherchons à l'estimer. Le raffinement ne cherche donc pas à estimer la valeur de la radiosité, mais l'erreur commise par l'estimation courante. Malgré l'importance de l'oracle, aucune heuristique valable dans tous les cas n'a encore été trouvée. Ce point est traité en détail dans le chapitre 4 page 69.

2.5 Regroupement de surfaces

Pour compléter la hiérarchie de subdivisions de surfaces, il est nécessaire de regrouper les surfaces initiales entre elles, récursivement, jusqu'à un groupe unique comportant la totalité de la scène. Ces groupes de surfaces, appelés *cellules* (*clusters*), vont permettre de calculer les échanges énergétiques à des niveaux de hiérarchies supérieurs aux surfaces.

Par exemple, calculer l'échange énergétique entre l'ensemble des surfaces composant une statue dans un hall d'opéra et une autre partie de ce hall peut-être grandement suffisant par rapport à l'approximation faite de la fonction de radiosité. Le *regroupement de surfaces* (*clustering*) a été introduit par Smits [SAG94] et Sillion [Sil94, Sil95] partant des études de hiérarchies de volumes afin d'accélérer les intersections de rayons en lançant de rayon.

Plusieurs structures de données peuvent être employées pour générer les cellules. Actuellement toutes sont basées sur des cellules parallélépipédiques dont les arêtes sont alignées avec les axes de la scène, mais il est imaginable d'utiliser des cellules quelconques.

On utilise en particulier deux structures dont la hiérarchie est décrite directement (pas de stockage nécessaire) :

- **Arbre octal uniforme** (*octree*) : L'espace de la scène est subdivisé en huit sous-parallélépipèdes de volumes égaux. Les calculs d'intersections sont très rapides car les contraintes spatiales sont fortes, mais ceci aux dépens de l'espace mémoire. Effectivement les objets de la scène se trouvant sur les plans de coupe de l'arbre impliquent la subdivision de la cellule [CLSS97].
- **Arbre binaire de partitionnement** (*kdtree*) : L'espace de la scène est subdivisé successivement par des plans parallèles aux axes, en deux sous espaces. Les calculs d'intersections sont moins rapides qu'en utilisant un arbre octal uniforme, mais l'espace mémoire utilisé est

moindre, parce que les boîtes "s'adaptent" aux objets de la scène de façon à minimiser le nombre de subdivision [Sil95].

Une structure dont la hiérarchie doit être explicitement décrite par des liens de parenté :

- **Cellule de regroupement hiérarchique** : C'est une boîte englobante dont les arêtes sont alignées avec les axes, qui établit des liens de parentés vers d'autres cellules qu'elle contient. Il existe plusieurs méthodes de création de ces cellules, et cela reste actuellement un problème ouvert[HDS99].

L'adjonction de la hiérarchie de cellules à la hiérarchie de surfaces permet d'obtenir une hiérarchie complète et ainsi une complexité en temps en $O(n \log n + k)$, avec n le nombre de surfaces initiales.

Une grosse difficulté induite par le regroupement intervient au moment du calcul de l'échange énergétique avec une cellule, le facteur de forme n'étant pas simple à exprimer. La valeur exacte du facteur de forme, dans des conditions de visibilité totale, entre une surface élémentaire dA_i et une cellule C_j composée de n surfaces $s_{1..n}$ est simplement la somme des facteurs de forme entre dA_i et chaque surface s_j incluant la visibilité interne à la cellule (F_{V_j}).

$$F_{dA_i, C_j} = \sum_{s_j \in C_j} \int_{p_j \in s_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} \mathcal{V}_j(p_i, p_j) ds_j$$

La fonction de visibilité interne \mathcal{V}_j (à ne pas confondre avec la fonction de visibilité externe à la cellule \mathcal{V}) peut-être fortement corrélée avec la direction de regard car les surfaces sont, par définition de la cellule, très proches. L'estimation courante est de supprimer cette fonction de visibilité ou au mieux de l'estimer par un coefficient d'extinction à l'intérieur de la cellule [Sil94]. Ces approximations sont très fortes et créent des artefacts visuels et énergétiques, hélas c'est le seul moyen actuel pour ne pas atteindre une complexité en $O(n^2)$ et donc traiter des scènes de taille importante. Les surfaces et les cellules possèdent des propriétés communes telles que : la possibilité de se subdiviser, la représentation simple de leur fonction de radiosité, etc... Ils sont appelés *éléments hiérarchiques*.

2.6 Radiosité par ondelettes.

La radiosité par ondelettes est une combinaison de la radiosité hiérarchique (représentation de la fonction de radiosité à différents niveaux de hiérarchie) et la radiosité de Galerkin avec une base de fonctions d'ordre supérieur à 0 (cf. §2.1, page 31). Les différentes bases de fonctions introduites par Peter Schröder et Steven Gortler pour la radiosité hiérarchique associent l'avantage de la hiérarchie, la facilité de conserver la cohérence entre les différents niveaux et l'ordre élevé des fonctions de la base qui permet une meilleure représentation de la fonction de radiosité [GSCH93, SGCH93].

La *radiosité par ondelettes* utilise les *ondelettes*, fonctions mathématiques découpant le signal entre valeur moyenne et détail. Une problématique engendrée par les ondelettes est la jonction entre les représentations de la fonction de radiosité sur les différentes surfaces. En effet, rien ne garantit la continuité entre des éléments hiérarchiques contigus. Pourtant, avec un choix approprié de points de quadrature, il est possible de minimiser les différences et d'obtenir une fonction quasiment continue [CAH00b]. La base de polynômes orthogonaux (par rapport au produit scalaire intégral sur $[-1, 1]$) couramment utilisée est construite à partir des polynômes de Legendre $P_n(x)$ de degré n .

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_{n+1}(x) &= \frac{(2n+1)xP_n(x) - nP_{n-1}(x)}{n+1} \end{aligned}$$

Actuellement les seuls ordres de base utilisés sont inférieurs ou égaux à 3. En dimension 0 (ondelettes de Haar), nous retrouvons la radiosité classique telle qu'elle fut initialement définie [HSA91].

Les ondelettes de degré 1 formalisent la *radiosité linéaire* (*vertex radiosity*) [MA92], et seules les ondelettes de degré 2 et 3 semblent utiles pour la représentation des détails de la fonction de radiosité.

L'algorithme de radiosité par ondelettes n'a jamais été réellement utilisé (notamment dans des produits commerciaux) car les temps de calculs, les problèmes de continuité et la place mémoire étaient complètement rédhibitoires tels que l'a montré Willmott dans une comparaison impartiale des différents algorithmes de radiosité [WH97]. Mais l'implémentation utilisée dans cette étude était la version initiale et brutale de l'algorithme de radiosité par ondelettes. Dernièrement, une modification de l'algorithme de base de la radiosité par ondelettes, supprimant le stockage des liens, choisissant les points de quadrature de façon adéquate et utilisant une parallélisation brutale a éliminé la majorité des problèmes de cette technique, la rendant accessible et très prometteuse quand aux résultats [CAH00a].

En utilisant cette nouvelle méthode, il semble que les ondelettes de degré supérieur deviennent intéressantes à partir d'une certaine qualité de solution désirée. Pour une faible qualité les ondelettes de Harr sont suffisantes et plus rapides, ensuite, plus on augmente la qualité, plus les ondelettes de degré supérieur deviennent intéressantes en temps de calcul [Cun00, HCA00].

3 Variations sur la radiosité.

Une majeure partie des travaux portant sur l'algorithme de radiosité tend à réduire le temps de calcul de la solution. Nous avons vu l'évolution de l'algorithme depuis sa formulation initiale, jusqu'à son état actuel.

Un ensemble d'optimisations plus ou moins indépendantes du type d'algorithme utilisé ont été développées. Ces optimisations cherchent principalement à diminuer le coût mémoire ou augmenter la qualité visuelle de la solution ou des images résultantes.

3.1 Réduction de mémoire.

Le calcul d'une solution de radiosité hiérarchique a initialement été décrit en stockant l'ensemble des liens (cf. §2.4, page 34). Chaque lien doit être connu pour chaque élément hiérarchique afin de pouvoir calculer, par collecte, l'irradiance arrivant sur cet élément. Le nombre de liens peut devenir très élevé en fonction de la taille de la scène, la taille des sources lumineuses et de la qualité de la solution désirée.

pour réduire les coûts mémoires, il est possible de supprimer le stockage des liens, en utilisant une méthode de propagation à la place de la méthode de collecte (cf. §2.3, page 31). Ainsi chaque "lien" propage sa puissance lumineuse instantanément dans la scène à chaque étape de calcul [SSSS98]. Hélas, la technique de collecte introduit de nouvelles difficultés telles que de trouver un critère de choix de l'émetteur et une convergence relativement faible.

La vitesse de convergence est lente due à la grande quantité de faibles émetteurs qui contribuent à l'éclairage indirect. Elle peut-être artificiellement accélérée en créant une solution initiale non nulle [Neu94, NTE95] ou en émettant plus d'énergie que lors d'un échange normal [FP92].

3.2 Parallélisation.

Les temps de calcul de solutions de radiosité sur des scènes complexes sont très importants malgré les récents algorithmes de radiosité hiérarchique ou radiosité par ondelettes. Il est naturel de paralléliser ces algorithmes afin de réduire les temps de calcul, un état de l'art sur la parallélisation des algorithmes d'éclairage global peut-être trouvé dans [CR98, RCJ98].

Il y a eu assez peu de travaux sur la parallélisation de la radiosité, en particulier, seule la radiosité hiérarchique a fait l'objet de telles études. Par contre, la radiosité stochastique entre dans le cadre de

la parallélisation des algorithmes basés sur Monte-Carlo. Les premiers essais de parallélisation de la radiosité hiérarchique furent relativement modestes avec un gain de 8,4 pour 64 processeurs [BG95]. Renambot *et al.* ont proposé un algorithme de parallélisation basé sur une approche géométrique, où chaque processeur effectue l'évaluation de la fonction de radiosité sur une petite partie de la scène. Les échanges énergétiques entre les différentes sous parties de la scène sont calculés par une technique d'interface virtuelle et de masque de visibilité [RAPP97]. Le gain de cette approche est plus important que toutes les précédentes, mais elle requiert une connaissance pointue de ses nombreux paramètres et demande une implémentation difficile.

Cavin a présenté une parallélisation d'un algorithme de radiosité par ondelette basé sur la technique de propagation (cf. §2.3, page 31) sur un nombre important de processeurs [CAP98]. La radiosité par ondelettes est plus précise mais aussi plus coûteuse en temps que la radiosité hiérarchique, d'où une nécessité de parallélisation. Les travaux de l'équipe ISA de l'INRIA de Nancy qui ont permis de rendre la radiosité par ondelette viable [CAH00b] se basent sur une parallélisation forte de leur algorithme [LAP00]. Les gains de temps sont importants (20 pour 24 processeurs dans le meilleur des cas), mais dépendent fortement du nombre de pipelines graphiques utilisés car leur implémentation en fait une utilisation importante durant les calculs de visibilité. Une étude précise de charge d'utilisation particulièrement adaptée pour le matériel *Origin2000*, montre un bon comportement jusqu'à 32 processeurs avec un gain d'accélération de 24, puis une chute importante et inexplicée des performances [Cav99].

Un autre type d'approche, basé sur une file d'attente des interactions, donne de très bons résultats au niveau des gains de temps. Singh place des paires d'éléments de surfaces dans une file d'attente par processeur. Quand une file est vide, le processeur inutilisé décharge les files d'attentes des autres processeurs. Les gains obtenus semblent excellents mais n'ont été évalués que sur une seule scène de taille très modeste [SGL94]. En ajoutant un répartiteur de tâches global pour les paires d'éléments hiérarchiques à traiter, les liens (cf. §2.4, page 34) en radiosité hiérarchique, et un processus de synchronisation des processeurs, Sillion *et al.* obtiennent de très bon résultats à moindres frais. Les gains sont variables selon les scènes, sans explication fournie, mais certains atteignent quasiment le gain maximal [SH00].

3.3 Radiosité dynamique.

Les images de synthèse sont souvent utilisées dans la création de scènes animées ; or la plupart des algorithmes ne sont adaptés qu'à la création d'instantanés, un film n'est alors qu'une succession d'images sans relations entre elles. Différentes approches ont été proposées pour intégrer la dimension temporelle dans le lancer de rayon [Gla88], en prenant en compte la cohérence temporelle en plus de la cohérence spatiale. S'il est alors possible de faire l'économie de multiples tests d'intersection, l'illumination est en revanche toujours calculée image par image.

L'algorithme de radiosité crée une solution énergétique indépendante du point de vue, permettant de l'utiliser interactivement dans un environnement virtuel, mais il est aussi fréquemment utilisé pour améliorer la qualité des images calculées par d'autres algorithmes (cf. §4.7, page 22). Un des axes de recherche est la réduction des calculs impliqués dans la mise à jour d'une solution déjà existante dont la géométrie de la scène est modifiée dynamiquement [BS96, ORDP96]. Une approche est d'effectuer des calculs de radiosité incrémentaux en propageant une correction énergétique dans le cadre de la radiosité progressive [Che90, GSG90, Geo90].

Une seconde approche, introduite par le développement de la radiosité hiérarchique, utilise les liens énergétiques, plus des considérations géométriques telles que les volumes de vision (*shaftculling*), afin de prédire les changements probables de la solution [FYT94, Sha97, DS97]. Pour des applications non interactives où les déplacements sont connus à l'avance, les recherches sont peu

nombreuses. Baum *et al.* calculent des facteurs de forme statiques et dynamiques autour des objets en mouvement [BWCG86]. Damez *et al.* génèrent une solution de radiosité en 4 dimensions en revenant sur l'hypothèse du régime temporel constant [DS99, Dam01].

3.4 Création d'images.

La solution de radiosité est indépendante du point de vue et peut donc être utilisée pour l'éclairage d'un monde virtuel. Cette solution n'est pas toujours d'une qualité visuelle excellente, principalement à cause des limites d'ombres difficilement repérées et rendues durant le calcul. Si l'on désire une image en temps non interactif, il est possible d'ajouter une passe de rendu qui va permettre de calculer une image d'une qualité supérieure. Cette passe est appelée *collecte terminale (final gathering)*.

A chaque point de l'image une estimation de la radiance est effectuée en fonction des liens qui arrivent sur l'élément hiérarchique correspondant. Les temps de calculs pouvant être très longs, des optimisations basées sur la cohérence spatiale ont été implémentées [Guo98, SSS01, SSP01].

L'amélioration de la qualité passe aussi par l'ajout de textures. La texture d'un matériau modifie sa BRDF, BRDF qui est considérée comme constante dans l'algorithme de radiosité. Il est possible de rajouter l'application de textures en post-traitement [CCWG88], mais la texture n'intervient alors pas dans le calcul de la solution. Pour intégrer la couleur de la texture dans la solution, il faut aussi prendre en compte la distribution de la réflectance [GSH94].

3.5 Contrôle d'erreur.

Si l'on arrive à quantifier les erreurs produites lors des échanges énergétiques, le contrôle de la qualité de la solution sera immédiat. Cyril Soler, durant sa thèse a montré qu'il est possible de calculer des bornes d'erreur entre deux surfaces dans le cas de visibilité totale [Sol98]. De nombreuses recherches portent sur le sujet du contrôle de l'erreur car ce serait alors une heuristique de raffinement fiable. Les bornes calculables sont, hélas, fréquemment larges [LSG94, SS96, BW96]. Certaines configurations géométriques permettent même de calculer les dérivées de la fonction de radiosité, obtenant ainsi une caractérisation précise de cette erreur [Hol96]. Malgré toutes les formes de bornes trouvées, la radiosité reste un algorithme itératif : les bornes s'éloignent à chaque itération, et deviennent très imprécises au bout de quelques itérations.

La visibilité est un problème non seulement ardu à déterminer, mais aussi très coûteux en temps. L'estimation de l'erreur produite dans l'interaction dû à la visibilité reste problématique et ne peut pas être borné facilement. Le complexe de visibilité [DDP97] caractérise toutes les configurations de visibilité possibles, mais cela reste d'une lourdeur inapplicable [DDP99].

Le regroupement provoque de nouvelles imprécisions en particulier à cause de la visibilité interne à la cellule. Si l'on considère la cellule comme une source uniforme d'énergie, l'erreur ne peut pas être bornée plus précisément qu'entre 0 et l'irradiance maximale, soit aucune information pertinente. Pour finir, les "facteurs de formes" (ou valeurs assimilées) impliquant une cellule sont tellement approximatifs que les bornes sont d'autant plus éloignées.

4 Conclusion

Il existe deux grandes classes d'algorithmes de résolution de l'équation de radiance, ceux basés sur des techniques stochastiques qui permettent de traiter des fonctions de réflectance complexes de façon approximative et souvent bruitée, et ceux basés sur une approche déterministe plus précise mais limitée à des surfaces lambertiennes.

L'algorithme de radiosité tente de résoudre le système d'équations intégrales d'interactions entre surfaces diffuses. Il donne des résultats physiquement réalistes mais ceci à des coûts de calculs et de mémoire importants. Les améliorations apportées permettent de traiter des scènes de plus en plus complexes, souvent avec une perte d'exactitude, comme la radiosité hiérarchique.

Le regroupement de surface simplifie le calcul des échanges énergétiques à un tel point qu'il devient difficile de l'estimer réellement. La charge de travail repose ensuite sur l'heuristique de raffinement qui doit estimer l'erreur probable induite dans l'échange. Une nouvelle approche utilise une hiérarchie incomplète et des bases de fonctions non constantes pour améliorer la représentation de la fonction de radiosité. Les résultats obtenus sont très encourageants malgré les temps de calculs importants.

La radiosité associée ou incorporée dans d'autres algorithmes de rendu calculant des échanges énergétiques de nature différentes permet des images photoréalistes d'une qualité visuelle aujourd'hui inégalée, mais elle souffre de problèmes de précision dus aux fortes simplifications effectuées afin de limiter les temps de calcul.

Avatars de surfaces

Les sens abusent souvent la raison par de fausses apparences.

Blaise Pascal

La fonction de radiosité d'une scène est unique à un instant donné ; elle est théoriquement indépendante de la description de la géométrie. Hélas, depuis la création de l'algorithme de radiosité, la géométrie joue un rôle important dans la qualité de l'approximation de la solution de radiosité calculée. Ceci pour deux raisons principales : la représentation de la fonction de radiosité par élément ou par groupe de surfaces et les différentes approximations effectuées lors du calcul de facteur de forme. La radiosité par ondelettes réduit les effets causés par la géométrie car la fonction de radiosité est représentée par une fonction plus complexe qu'une simple fonction constante et permet d'être plus indépendant du support, mais sans régler le problème définitivement. Afin de réduire cette dépendance à la représentation géométrique, nos recherches nous ont guidé vers une simplification de la géométrie en prétraitement du calcul de la solution.

Ces recherches sont parallèles et complémentaires aux travaux réalisés par l'équipe de l'INRIA de Nancy portant sur le *virtual meshing* [HCA00]. Après une étude de la problématique posée par la géométrie dans l'algorithme de radiosité, nous verrons une approche similaire au *virtual meshing* spécialisée dans les surfaces planes. Puis nous proposerons plusieurs algorithmes capables d'automatiser la phase de génération d'avatars ainsi qu'une étude de leurs paramètres.

1 Radiosité et modèles géométriques.

1.1 Génération de maillages.

Dans l'algorithme de résolution matricielle ou en radiosité progressive, la modélisation de la scène tient une place prépondérante dans la qualité de la solution car la géométrie n'est pas modifiée lors du calcul de la solution. Les limites d'ombres doivent être "prévues" par le créateur du modèle de la scène, ce qui implique l'impossibilité de modifier la position des sources lumineuses. Pour résoudre ce problème, deux méthodes ont été mises au point : la hiérarchie de surface (cf. §2.4, page 33) et le maillage de discontinuité. Quand bien même elles limitent la dépendance entre la solution finale et la modélisation initiale, il est toujours nécessaire d'avoir une modélisation particulière composée des surfaces les plus grandes possibles. Ce type de modélisation est plus simple pour un graphiste mais il n'est pas en accord avec les logiciels actuels qui préfèrent un grand nombre de polygones simples (généralement des triangles) à un petit nombre de polygones complexes (fig. 3.1).

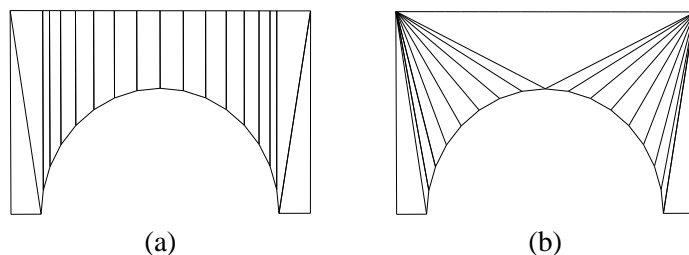


FIG. 3.1: Triangulation d'une arche par deux logiciels courants de modélisation (a) blender (b) 3D Studio Max.

Le regroupement de surfaces (clustering) est une solution qui pourrait sembler suffisante pour régler ce problème en calculant les échanges énergétiques directement sur l'ensemble de surfaces. Hélas, une grande quantité d'informations intéressantes est perdue lors de la phase de regroupement, dont la notion de coplanarité. C'est pourquoi nous cherchons à étendre la notion de groupe de surfaces, en dérivant des groupes (clusters) ayant des propriétés spécifiques desquelles nous pouvons tirer avantage.

1.2 Approximation du facteur de forme.

La seconde source de dépendance entre la géométrie de la scène et la solution de radiosité se situe au niveau du calcul de facteur de forme. Il y a une littérature abondante sur le calcul de facteur de forme (cf. §1.2, page 27) mais dans la majorité des configurations, il n'existe pas une formulation exacte : on utilise diverses estimations. Les approximations induites sont d'autant plus importantes que les polygones sont dégénérés. Dans l'exemple de la figure 3.1 la somme des facteurs de formes peut être très différente, les polygones de la figure (b) étant beaucoup plus dégénérés. Pour réduire les erreurs commises dans ces configurations, il existe plusieurs types de solutions :

- Sur-échantillonnage : par une simple méthode de Monte-Carlo, calculer le facteur de forme grâce à un ensemble d'échantillons point-points. Ceci prend en compte la totalité des configurations géométriques polygone-polygone, mais est énormément coûteux en temps de calcul.
- Simplification de la géométrie : un polygone dégénéré est souvent un élément d'une surface plus complexe. On peut calculer le facteur de forme d'un polygone simplifié et le reporter sur les petits polygones complexes qui le composent.

La modélisation de formes arrondies est source d'un grand nombre de surfaces polygonales dégénérées. L'exemple suivant modélise un double engrenage, les cercles centraux ont engendré une multitude de polygones allongés (fig. 3.2a) dont l'approximation de facteurs de formes donne des résultats relativement mauvais (fig. 3.2b).

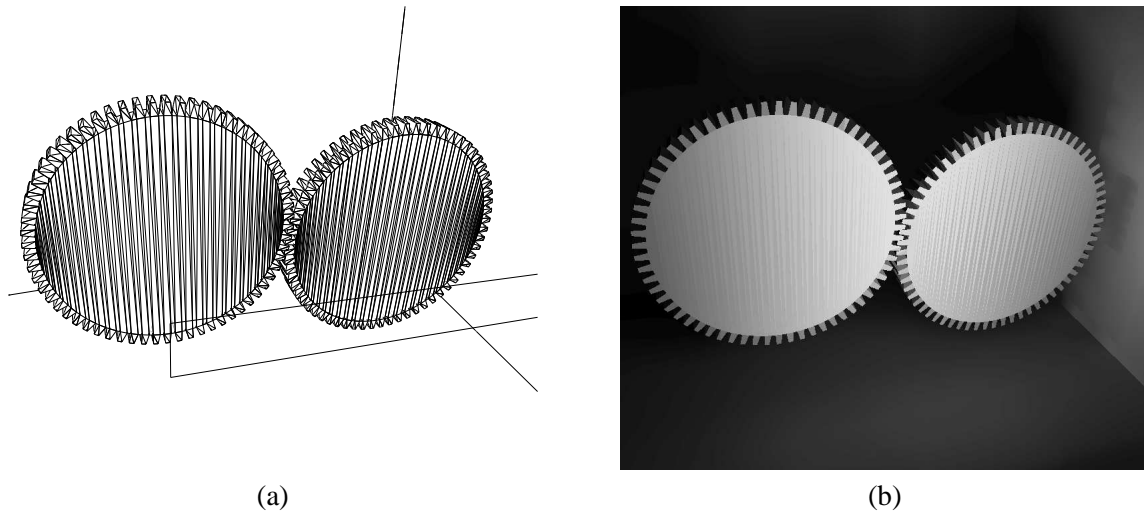


FIG. 3.2: Approximations des facteurs de formes dans un cas de maillage dégénéré. (a) Maillage de la scène. (b) Artéfacts visuels engendrés.

2 Avatars de surfaces.

Nous proposons un type particulier de regroupement de surfaces dans lequel on connaît des informations supplémentaires par rapport à un groupe de surfaces classique : les *avatars de surface*. Dans notre cas, la principale propriété que nous voulons conserver est la notion de coplanarité. Un avatar, en tant que regroupement de surfaces coplanaires, est un polygone particulier qui n'est pas forcément convexe ou connexe.

Les informations supplémentaires permettent de réduire la perte d'informations très importante lors du passage de surfaces à regroupement de surfaces. Par exemple, dans le cas du clavier du terminal présenté dans la figure 3.3, la connaissance de coplanarité de l'ensemble des touches permet une meilleure approximation des échanges énergétiques que dans le cas d'un regroupement classique (fig. 3.3a) car les groupes sont mieux répartis (fig. 3.3b).

Un avatar de surfaces est une surface englobant un ensemble d'autres surfaces dans le but de simplifier et de diminuer la quantité de calculs et d'augmenter leur précision, principalement pour les calculs de visibilité et de facteurs de forme. Les avatars de surfaces ont été définis pour des surfaces paramétriques quelconques [Cun00, HCA00], mais leur génération est actuellement réalisée en prétraitement manuel – ce qui est un problème de modélisation beaucoup plus compliqué que le problème initial à résoudre –. Il est nécessaire de trouver un algorithme de génération automatique des avatars de surfaces pour les rendre utilisables. Une approche, légèrement différente des avatars de surfaces, cherche à créer une hiérarchie de surfaces, représentant un ensemble de surfaces à peu près coplanaires [WHG99]. Cette technique n'est pas applicable dans notre cas, car elle repose sur la simplification de la géométrie avec perte de connaissance d'un niveau à l'autre de la hiérarchie de

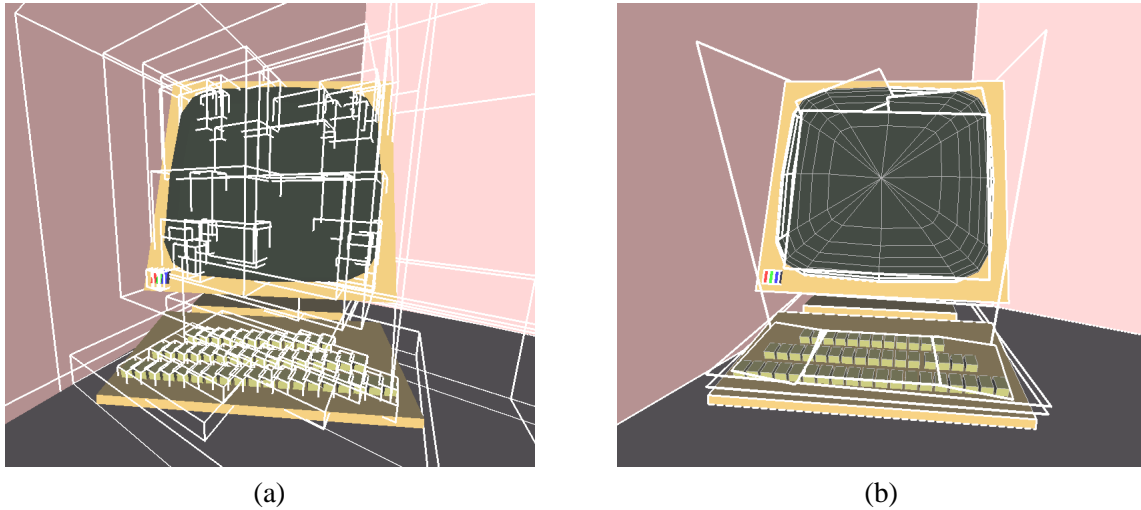


FIG. 3.3: Création de groupes de surfaces ayant une connaissance supérieure aux groupes de surfaces classiques. (a) Pas de connaissance à priori. (b) Connaissance de coplanarité.

groupement de surfaces (*face cluster*). Or nous ne désirons pas créer une hiérarchie et surtout ne pas perdre des informations lors de la simplification de la géométrie.

Après une description détaillée des avatars de surfaces plans, nous proposons dans les paragraphes suivants trois algorithmes de génération automatique, ainsi qu'une étude des différents paramètres qui régissent ces algorithmes.

2.1 Avatars de surfaces plans.

Les avatars de surfaces plans sont un sous-ensemble des avatars de surfaces paramétriques définis dans [HCA00]. Nous nous limitons à ce cas simplifié car la génération automatique est un problème suffisamment complexe pour l'attaquer pas à pas. Nous recherchons, par les avatars de surfaces plans, à remplacer des surfaces *quasi-coplanaires* par une surface qui représentera l'ensemble. Nous précisons la notion de *quasi-coplanarité* (cf. §4, page 51).

Représentation géométrique.

Un avatar de surface plan est un polygone quelconque, préférablement simple (quadrilatère ou triangle) de façon à obtenir des estimations les plus précises possible lorsqu'il intervient dans les calculs. Ce polygone est nommé *polygone-support*, il est un englobant d'un ensemble de polygones presque coplanaires appelés *polygones internes*, ceux-ci ne sont pas forcément jointifs ou disjoints. Les avatars sont des éléments de surfaces hiérarchiques. Ils peuvent donc être subdivisés en plusieurs sous-avatars de surfaces ayant des caractéristiques similaires. L'avatar de surfaces est utilisé comme s'il s'agissait d'un simple polygone sauf lors de :

- la méthode d'échantillonnage ;
- le calcul de facteur de forme ;
- les calculs de visibilité ;
- la visualisation ;
- le push/pull.

Nous définissons deux caractéristiques propres aux avatars :

- Le taux de remplissage : c'est la quantité relative de surface réellement utile de l'avatar. Le taux de remplissage se définit simplement comme le rapport entre la somme des aires des polygones internes et l'aire du polygone support

$$\tau = \frac{\sum_i \|\mathcal{P}_i\|}{\|\mathcal{A}\|}, \text{ avec } \|\mathcal{P}_i\| \text{ l'aire du polygone interne } i \text{ et } \|\mathcal{A}\| \text{ l'aire de l'avatar.}$$

Pour des polygones internes dont les projections sont disjointes, cette valeur est comprise entre 0 et 1 et reflète exactement la quantité de surface du polygone-support utile. Si les polygones se chevauchent, ce taux de remplissage peut dépasser 1. Nous verrons que cette valeur nous est utile dans le push/pull.

- La régularité : ce ratio décrit le "degré de dégénérescence" d'un polygone, c'est-à-dire s'il est plus ou moins proche d'un polygone régulier. Pour un polygone quelconque nous avons utilisé le rapport des longueurs entre la plus petite et la plus grande arête. C'est une évaluation de la régularité relativement précaire mais qui donne des résultats satisfaisants dans notre cas. Dans le cadre de cette étude, nous nous sommes cantonnés à des polygones simples tels que triangles et rectangles.

Les prochaines sections définissent les caractéristiques précédemment citées qui diffèrent des polygones classiques.

Échantillonnage.

Les échantillons doivent se situer sur les polygones internes et non pas dans les zones "vides" de l'avatar. Dans le cas d'un tirage aléatoire, un simple test d'appartenance des points dans les polygones internes est suffisant. Par contre, dans le cas d'un nombre défini d'échantillons répartis uniformément, le problème revient à définir un échantillonnage uniforme d'un polygone quelconque. Une méthode statistique simple utilise un tirage uniforme sur le polygone support de l'avatar en divisant le nombre d'échantillons par le taux de remplissage de l'avatar et en ne gardant que les points se trouvant dans un polygone interne. Si l'ensemble des polygones internes n'est pas trop dégénéré et le nombre d'échantillon désiré pas trop petit, le nombre de points tirés sera proche du nombre souhaité, mais rien ne le garantit.

Dans l'exemple suivant, on désire obtenir 85 échantillons de l'avatar. Le taux de remplissage est de 0,43. Le nombre d'échantillons au niveau de l'avatar est 198 (85/0,43) réduit à $14^2 = 196$ pour créer un échantillonnage régulier du carré formé par l'avatar. Sur les 196, 83 échantillons appartiennent aux surfaces internes (fig. 3.4).

Calcul de facteur de forme.

Le calcul du facteur de forme par échantillonnage pose uniquement le problème de l'échantillonnage de l'avatar discuté précédemment. Une approche simple est de calculer une moyenne des facteurs de formes point-récepteur en ne prenant en compte que le taux de remplissage global du support.

$$F_{pq} = \tau_p \frac{\sum_{P_n \in p} F_{P_n q}}{N} \quad (3.1)$$

avec P_n le n ème échantillon du polygone p et $F_{P_n q}$ le facteur de forme point-polygone obtenu par une méthode quelconque.

Une méthode plus précise mais beaucoup plus coûteuse est de sommer récursivement l'ensemble des facteurs de formes des subdivisions de l'avatar. Ainsi, les pondérations par les taux de remplis-

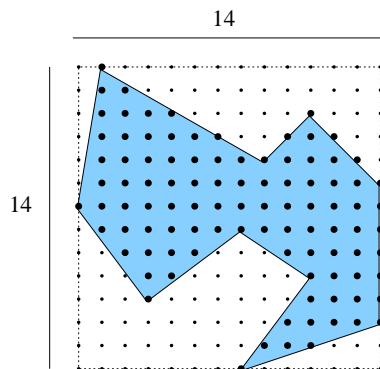


FIG. 3.4: Échantillonnage d'un polygone quelconque par échantillonnage de son polygone englobant.

sages sont considérées non pas sur l'ensemble du support mais sur chaque subdivision de façon à affiner le résultat.

Si p n'a pas de subdivision : F_{pq} est calculé comme précédemment (eq. 3.1).
 sinon p est subdivisé en S subdivisions : $F_{pq} = \sum_{s=1}^S F_{p_s,q}$

Visibilité.

La prise en compte de la visibilité lorsque l'avatar est considéré comme un obstacle potentiel doit se faire avec la géométrie initiale de façon à reproduire précisément les ombres de la scène. Si une intersection existe entre un rayon et l'ensemble des polygones internes, alors cette intersection est la même que l'intersection de ce rayon et du polygone support. La méthode la plus simple est de trouver le point d'intersection avec le polygone support puis de vérifier s'il appartient à au moins un des polygones internes. Cette détermination de l'appartenance d'un point dans un polygone en deux dimensions est largement décrite dans différents domaines de l'imagerie numérique [Hec94, MT97].

Dans la figure (fig. 3.5), les polygones exactement coplanaires du mur central ont été placés dans un avatar. On peut remarquer le passage de la lumière au travers de la porte due à la gestion de la visibilité utilisant la géométrie initiale.

Visualisation.

Deux solutions sont envisageables pour la visualisation. La première méthode est d'afficher l'avatar en masquant ce qui ne doit pas être visible, option gérée aisément par les pochoirs (*stencil buffers*) d'Open GL* (algo. 3.1). Les temps d'affichage sont très légèrement supérieurs à la visualisation de la géométrie brute mais restent interactifs.

Dans la figure (fig. 3.5b), l'utilisation des pochoirs permet de démasquer la porte éliminée de la géométrie par les avatars.

Avec la visualisation basée sur les pochoirs, certains petits artefacts apparaissent lorsque les avatars sont visualisés avec un angle rasant. Les pochoirs sont exactement planaires alors que les avatars sont simplement quasi-plans, ainsi les surfaces qui n'appartiennent pas exactement au plan de l'avatar sont projetées à l'extérieur de son polygone support pour des angles éloignés de la normale à l'avatar (fig. 3.6).

*bibliothèque standardisée d'affichage graphique

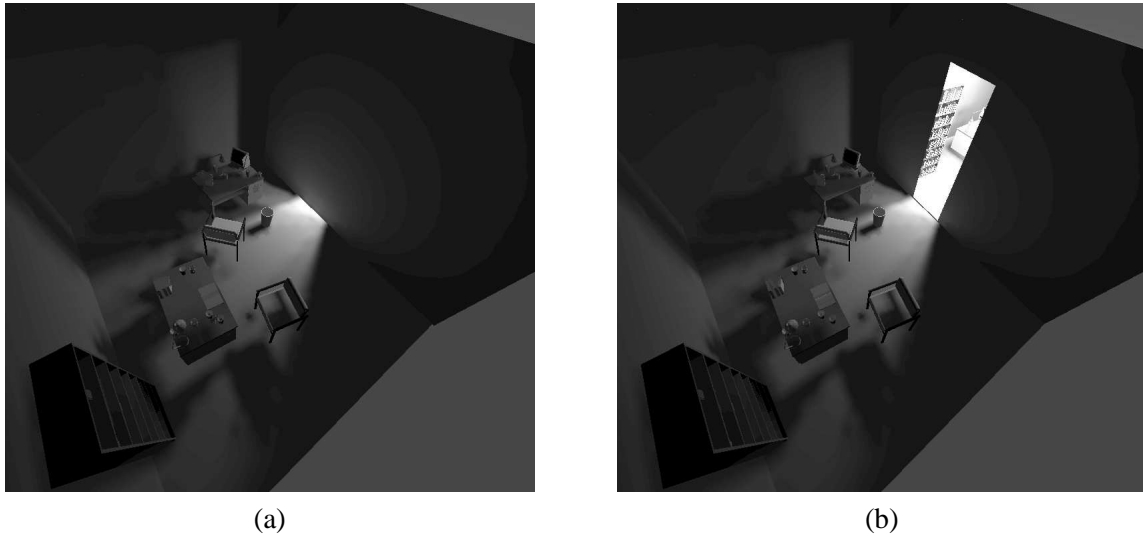


FIG. 3.5: Visualisation des avatars en utilisant les pochoirs d'Open GL. (a) Géométrie initiale des avatars. (b) Ajout de pochoirs afin d'éliminer les zones invisibles.

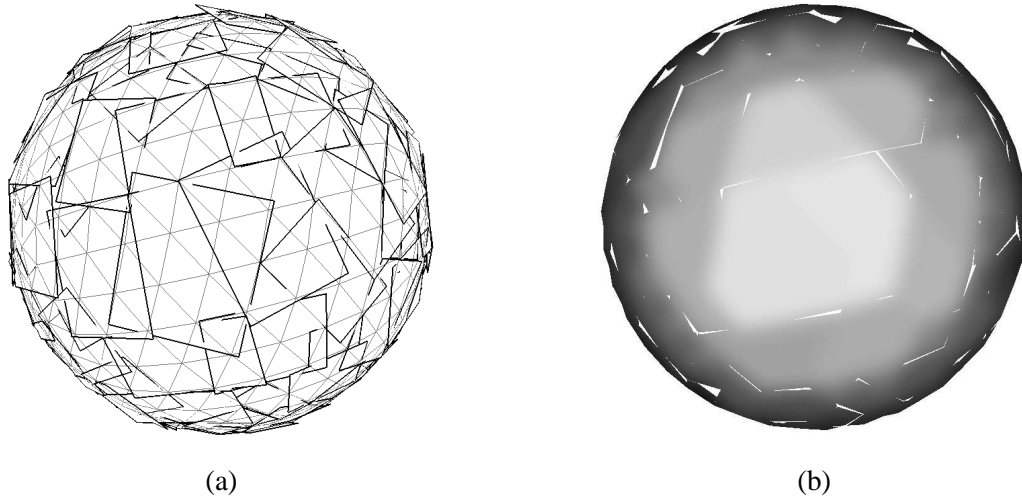


FIG. 3.6: Artefacts de visualisation générés par l'utilisation des pochoirs. (a) Avatars générés sur une sphère. (b) visualisation par pochoirs.

Le second type de visualisation permet d'éliminer ces artefacts. Il se base sur le calcul des valeurs de radiosité des polygones internes à partir de la fonction de radiosité de l'avatar, ceci en post-traitement. L'avatar possède une représentation de la fonction de radiosité sur son support. On peut alors calculer, pour chaque sommet des polygones internes, la valeur de radiosité de la fonction du polygone support. Pour chaque polygone, on calcule sa valeur de radiosité en faisant la moyenne des radiosités à ses sommets. Ainsi, il est possible d'afficher exactement la géométrie initiale sans aucun artefacts.

Algorithme 3.1 Visualisation des avatars plans avec le stencil buffer d'Open GL

```

1 - Initialise le contexte graphique
glClearStencil(0)
glClear(GL_STENCIL_BUFFER_BIT)           Initialise le Stencil Buffer
2 - Dessine le masque dans le stencil buffer
glEnable(GL_STENCIL_TEST)                Autorise l'utilisation du stencil buffer
glDisable(GL_DEPTH_TEST)                 Désactive le test du tampon de profondeur
glStencilMask( 0)                         Limite l'utilisation du Stencil à 1 bit (vrai-faux)
glColorMask(0,0,0,0)                     Désactive l'utilisation du tampon de dessin
glStencilFunc(GL_ALWAYS,1, 0)             Place les bits du stencil buffer à 1 sans conditions
glStencilOp(GL_KEEP,GL_KEEP,GL_REPLACE)
    DrawInternalPolygons()                Affiche l'ensemble des polygones internes
3 - Dessine à l'avatar à l'intérieur du masque
glEnable(GL_DEPTH_TEST)                  Réactive l'utilisation du tampon de profondeur
glColorMask(1,1,1,1)                    Réactive le tampon de dessin pour modifier l'image finale
glStencilFunc(GL_EQUAL,1, 0)              Ne dessine que si le stencil buffer est égal à 1
glStencilOp(GL_KEEP,GL_KEEP,GL_KEEP)
    DrawSupportPolygon()                  Affiche le polygone support entier
glDisable(GL_STENCIL_TEST)                Désactive l'utilisation du stencil buffer

```

Push-Pull.

Les avatars de surfaces possèdent sur leur polygone-support plus d'énergie que l'ensemble des polygones internes. Il ne faut donc pas distribuer toute l'irradiance en bas de la hiérarchie mais seulement une partie afin de conserver la cohérence énergétique. L'irradiance est calculée par unité d'aire ; la valeur à distribuer aux fils (lors du *push*) est donc simplement la valeur de l'irradiance de l'avatar. Par contre, il est d'usage de moyenniser les valeurs d'énergie lors de la remontée dans la hiérarchie (*pull*), ceci afin de pouvoir prendre en compte les objets ponctuels. L'énergie étant dépendante de la taille du polygone, il faut prendre en compte le taux de remplissage de chaque fils pour le père, cette valeur d'énergie est d'abord multipliée par le taux de remplissage avant de moyenniser les valeurs de l'ensemble des fils. Les polygones internes pouvant se chevaucher, le taux de remplissage peut dépasser sa valeur maximale théorique de 1. Il est important de ne pas la tronquer afin de ne pas perdre d'énergie d'un niveau à l'autre de la hiérarchie.

2.2 Création de la description géométrique.

Les avatars de surfaces dérivant des surfaces classiques, il faut définir leur description géométrique :

1. Trouver le plan du polygone-support.
2. Trouver la forme du polygone-support.

1 - Détermination du plan.

Les surfaces étant relativement proches les unes des autres, une simple moyenne pondérée des polygones internes suffit grandement pour déterminer la plan du polygone support. Nous avons utilisé l'aire de chaque polygone pour refléter son importance.

$$\vec{\mathcal{A}}_n = \frac{\sum_{i=1}^N \vec{\mathcal{P}}_{n_i} \|\mathcal{P}_i\|}{\sum_{i=1}^N \|\mathcal{P}_i\|}, \quad \mathcal{A}_c = \frac{\sum_{i=1}^N \mathcal{P}_{c_i} \|\mathcal{P}_i\|}{\sum_{i=1}^N \|\mathcal{P}_i\|}$$

où : $\vec{\mathcal{A}}_n$ et \mathcal{A}_c sont respectivement le vecteur normal et le centre de l'avatar, ainsi que $\vec{\mathcal{P}}_{n_i}$ et \mathcal{P}_{c_i} pour le polygone interne \mathcal{P}_i . $\|\mathcal{P}_i\|$ représente l'aire du polygone.

2 - Détermination de la forme.

La première étape est de projeter l'ensemble des polygones internes sur le plan support. Par définition, les polygones internes sont quasi-coplanaires, leurs projections sont très proches de leurs descriptions dans l'espace. Nous n'avons besoin que de projeter les sommets des polygones pour obtenir les informations nécessaires car nous allons chercher un englobant de l'ensemble des polygones.

L'étape suivante est de trouver un polygone bidimensionnel englobant l'ensemble des points projetés dans le plan support, l'aire de ce polygone doit être la plus petite possible. La forme du polygone englobant dépend du type d'avatar à générer, les plus simples étant des triangles ou des rectangles. Les schémas présentés se basent sur le cas le plus simple : les rectangles.

Trouver le rectangle minimal est un problème résolu par la technique des *droites d'appuis tournantes (rotating calipers)*, mais nécessite de connaître l'enveloppe convexe de l'ensemble des points [Tou83]. Nous avons utilisé une méthode simple et efficace qui permet de trouver précisément le meilleur candidat sans avoir besoin de trouver l'enveloppe convexe. Cet algorithme se décompose en deux étapes :

- **Échantillonnage** : on découpe l'ensemble des directions de 0 à $\frac{\pi}{2}$ en n échantillons. A chaque direction α_i on calcule le rectangle englobant $Eng(\alpha_i)$ en fonction de la rotation correspondante (fig. 3.7a). On repère l'indice minimal i_{min} tel que $\forall i \neq i_{min}, \quad \|Eng(i)\| > \|Eng(i_{min})\|$.
- **Dichotomie** : Si l'échantillonnage précédent n'est pas trop grossier, il est fortement probable que l'angle correspondant à l'aire minimale soit compris entre $\alpha_{i_{min}-1}$ et $\alpha_{i_{min}+1}$. La fonction reliant l'aire du rectangle à l'angle est considérée monotone sur cet intervalle ; nous pouvons ainsi affiner le résultat en utilisant une recherche de minimum par dichotomie autour de l'angle minimal trouvé par échantillonnage (fig. 3.7b).

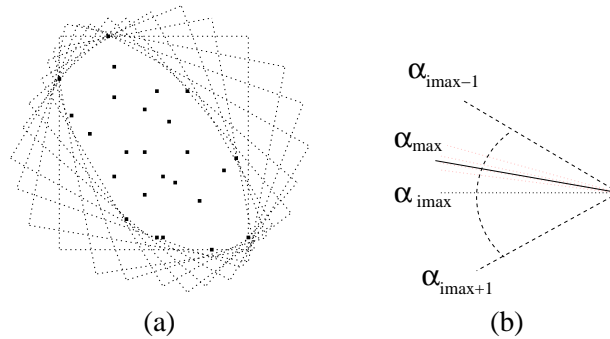


FIG. 3.7: Génération de l'enveloppe convexe rectangulaire d'aire minimale en deux étapes : (a) trouver le meilleur candidat par échantillonnage, (b) affiner la solution par dichotomie.

3 Génération automatique d'avatars de surfaces plans.

Nous voulons trouver un ou plusieurs polygones qui simplifient la géométrie de la scène par le regroupement de polygones quasi-coplanaires. Les données en entrée sont : un ensemble de surfaces quelconques (la géométrie de la scène). La sortie doit être : un ensemble d'*avatars de surfaces* (polygones qui regroupent un sous-ensemble des polygones de la scène).

Nous verrons la complexité du problème ainsi que la subdivision en deux étapes que nous utilisons afin de la réduire. Puis, après avoir défini nos critères désirés de génération, nous décrirons en détail, dans les sections 4 et 5, les deux différentes étapes.

3.1 Réduction de la complexité.

Soit n le nombre de surfaces initiales et m le nombre de surfaces dans chaque avatar ($m < n$). Dans le pire des cas, il faut trouver les ensembles quasi-coplanaires dans la scène et toutes les combinaisons possibles $O(n!)$, puis l'ensemble des avatars dans ces ensembles $O(m!)$, la complexité globale atteint $(m! \cdot n!)$. Cette complexité est très approximative, mais le nombre de solutions n'est pas à ce jour (ni dans un futur proche) calculable en un temps raisonnable. Afin de réduire la complexité de cette recherche, nous proposons de découper le problème en deux sous-problèmes (fig. 3.8) :

- Trouver un ensemble de surfaces quasi-coplanaires dans la géométrie de la scène.
- Trouver un ensemble d'avatars dans un ensemble de surfaces quasi-coplanaires.

Cette division en deux étapes les rend faussement indépendantes, ceci empêche de faire dépendre la notion de quasi-coplanarité des propriétés de l'avatar. Par exemple, il n'est pas possible d'autoriser le regroupement de surfaces faiblement coplanaires sur un petit avatar et de n'autoriser que des surfaces fortement coplanaires sur un avatar de taille importante.

De plus l'ensemble des surfaces quasi-coplanaires n'est pas unique ; il faut trouver un ensemble satisfaisant sans prendre en compte les futurs avatars à créer.

Par cette division du problème nous limitons de façon importante les choix possibles parmi l'ensemble des solutions, mais c'est la plus naturelle afin de réduire la complexité de cette tâche.

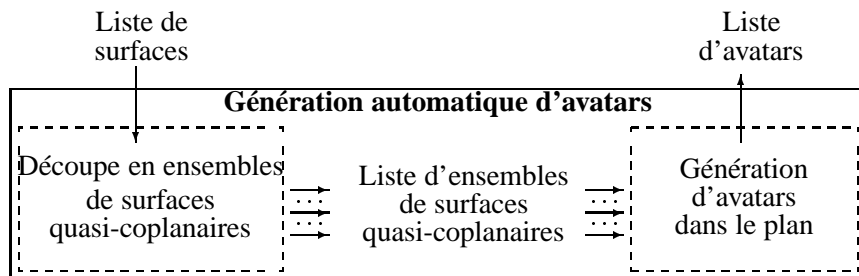


FIG. 3.8: Découpe du problème de génération automatique d'avatars en sous-problèmes.

3.2 Critères de génération.

Afin de ne pas trop compliquer la génération d'avatars, nous ne définissons que deux critères :

- Maximisation du taux de remplissage.
- Maximisation du nombre de polygones internes.

Ces deux critères sont antagonistes ou du moins ne vont pas dans le même sens de recherche de la solution. Si l'on ne considérait que la maximisation du taux de remplissage, il suffirait de créer un seul avatar par polygone. Si l'on ne considérait que la maximisation du nombre de polygones internes, le

problème serait simplifié à "chercher les ensembles de polygones quasi-coplanaires" et créer un seul avatar.

La difficulté de la génération d'avatars se situe dans la recherche d'une solution acceptable dans la fonction qui relie un ensemble d'avatars à la géométrie de la scène.

4 Regroupement en surfaces quasi-coplanaires.

Le but de la première partie de l'algorithme est de trouver un ensemble de surfaces quasi-coplanaires dans la scène. Le critère de coplanarité doit être défini de manière à regrouper des surfaces relativement proches en inclinaison et en "différence de hauteur". (fig. 3.9).

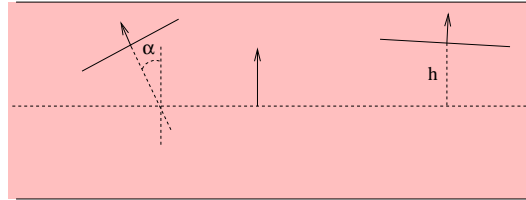


FIG. 3.9: Critère de coplanarité entre surfaces dépendant de l'angle α et de la distance h .

Une fois la coplanarité entre surfaces définie par un opérateur, nous verrons 3 algorithmes de regroupement de surfaces coplanaires ayant des caractéristiques différentes en temps et mémoire.

4.1 Opérateur de coplanarité.

La coplanarité est une mesure de la distance entre une surface et un avatar ; il n'existe pas de définition normalisée de cette mesure. L'avatar peut être vide au moment où l'on commence à le créer ; dans ce cas, la valeur de la distance avec n'importe quelle surface est nulle.

Nous définissons la distance comme un couple de valeurs, la première calculant l'angle entre les normales et la seconde, la distance du centre du polygone au plan-support de l'avatar (eq. 3.2). Cette définition a l'avantage de prendre en compte implicitement la taille de la surface dans la seconde valeur du couple. Ceci élimine les surfaces éloignées à cause de leur taille excessive.

$$\begin{aligned}
 d_{\mathcal{A}} : Avatar \times Polygone &\longrightarrow \mathbb{R} \times \mathbb{R}^+ \\
 (\emptyset, \mathcal{P}) &\longmapsto (0, 0) \\
 (\mathcal{A}, \mathcal{P}) &\longmapsto (d_{\mathcal{A}}^a, d_{\mathcal{A}}^h) = \left(\widehat{\vec{\mathcal{A}}_n, \vec{\mathcal{P}}_n}, \left| \frac{\vec{\mathcal{A}}_c \cdot \vec{\mathcal{P}}_c \cdot \vec{\mathcal{A}}_n}{\|\vec{\mathcal{A}}_n\|} \right| \right)
 \end{aligned} \tag{3.2}$$

$$\text{Avec } \begin{cases} \vec{\mathcal{P}}_n & \text{La normale à la surface } \mathcal{P}, & \vec{\mathcal{A}}_n & \text{La normale à l'avatar } \mathcal{A}. \\ \mathcal{P}_c & \text{Le centre de la surface } \mathcal{P}, & \mathcal{A}_c & \text{Le centre de l'avatar } \mathcal{A}. \end{cases}$$

On dérive de cette distance un opérateur binaire de coplanarité qui détermine si un polygone est coplanaire à un avatar. Soit \asymp l'opérateur de coplanarité.

$$\begin{aligned}
 \asymp : Avatar \times Polygone &\longrightarrow \mathbb{B} \\
 (\mathcal{A}, \mathcal{P}) &\longmapsto |d_{\mathcal{A}}^a| < \varepsilon_a \wedge d_{\mathcal{A}}^h < \varepsilon_h
 \end{aligned} \tag{3.3}$$

Où $(\varepsilon_a, \varepsilon_h)$ sont deux seuils spécifiés par l'utilisateur appartenant à $\mathbb{R} \times \mathbb{R}^+$.

4.2 Algorithme glouton.

L'algorithme glouton de regroupement correspond à la partie "classification" dans le schéma de l'algorithme complet de génération (fig. 3.10). Il consiste à prendre tous les polygones de la scène les uns après les autres dans un ordre quelconque en essayant de les intégrer dans le groupe de surfaces coplanaires courant, le premier polygone étant toujours intégré. La complexité en temps est inférieure à $\Theta(n^2)$ car les polygones agrégés ne seront pas testés ultérieurement, mais elle reste en $O(n^2)$.

L'ordre de traitement est important et peut modifier les groupes trouvés car la première surface traitée donne la direction principale du groupe. Cette approche n'est valable que pour des scènes de petite taille. (algo. 3.2).

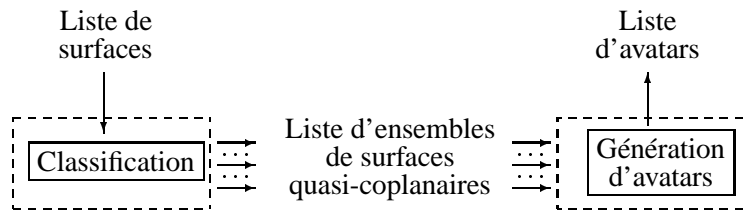


FIG. 3.10: Création d'avatar par un algorithme glouton

Algorithme 3.2 Regroupement en surfaces quasi coplanaires glouton

Regroupement Glouton sur $\{S\}$

$\{S\}$ un ensemble de surfaces

```

pour toutes les surfaces  $S$  non utilisées de  $\{S\}$  faire
    créer un ensemble de surfaces quasi-coplanaires  $Q$  contenant  $S$ 
    noter  $S$  comme surface utilisée
    pour toutes les surfaces  $s$  non utilisées de  $\{S\}$  faire
        si ( $S \asymp s$  est vrai) alors
            ajouter  $s$  à  $Q$ 
        fin si
    fin pour
    si ( $\|Q\| \geq 1$ ) alors
        ajouter  $Q$  à  $\{Q\}$ 
    fin si
fin pour

retourner  $\{Q\}$ 
  
```

4.3 Regroupement par hiérarchie.

On trouve dans la littérature scientifique (particulièrement en géométrie algorithmique) une technique classique de réduction de la complexité reposant sur le vieil adage *diviser pour régner*. Le principe général du traitement d'une tâche est de morceler une tâche complexe en sous-tâches, d'effectuer un traitement plus simple sur ces sous-tâches et de fusionner le résultat des traitements.

Dans le cadre de la génération d'avatars, nous n'allons pas utiliser l'ensemble des surfaces de la scène mais plusieurs sous-ensembles. Les avatars créés seront ensuite regroupés. Les avatars ayant plus de chances d'être regroupés géographiquement, il semble naturel de regrouper les surfaces en

sous-ensembles "proches", ce qui peut être aisément réalisé grâce à la hiérarchie déjà créée par l'algorithme de "regroupement de surfaces" (*clustering*). Si l'on désire regrouper les surfaces en sous-ensembles de, au plus, N surfaces, l'algorithme suivant peut-être utilisé (algo. 3.3).

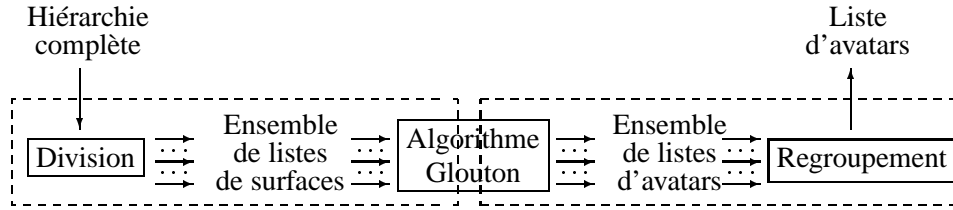


FIG. 3.11: Création d'avatars en utilisant la hiérarchie.

Algorithme 3.3 Regroupement en surfaces quasi coplanaires grâce à la hiérarchie

Regroupement Par Hiérarchie sur \mathcal{G}

\mathcal{G} un groupe de surfaces (*cluster*)

si ($\|\mathcal{G}\| < N$) **alors**

$\{\mathcal{A}\} \leftarrow$ l'ensemble des avatars créés par **Regroupement Glouton**(\mathcal{G})

sinon

pour tous les fils G_i de \mathcal{G} **faire**

$\mathcal{A}_i \leftarrow$ les avatars créés par **Regroupement Par Hiérarchie**(G_i)

fin pour

$\{\mathcal{A}\} \leftarrow$ l'ensemble des avatars créés par **Regroupement d'Avatars**($\{\mathcal{A}_i\}$)

fin si

retourner $\{\mathcal{A}\}$

L'algorithme présenté effectue, en fait, les deux étapes générales de la génération d'avatars, et non pas seulement la découpe en ensembles de surfaces quasi-coplanaires (fig. 3.11). Le parcours de la hiérarchie de clusters correspond au regroupement en ensembles quasi-coplanaires, alors que la remontée correspond à la collecte des avatars créés à l'aide d'un algorithme glouton complet. La génération d'avatars est, en fait, prise en compte par celui-ci.

4.4 Regroupement par espace dual.

Le regroupement par espace dual correspond aux tâches de "transformation de l'espace dual", "clusterisation de l'espace dual" et "classification" dans la figure (fig. 3.12).

Le but de la transformation par espace dual est de pouvoir définir une distance de coplanarité entre surfaces qui soit monotone. Ceci afin de pouvoir utiliser des techniques d'optimisation (kd-tree, BSP, grilles régulières, boîtes englobantes, etc...) permettant une recherche rapide des plans les plus "proches" au sens de la coplanarité. La distance de coplanarité telle que nous l'avons décrite (eq. 3.2), se base sur l'angle entre les surfaces. Il nous faut une transformation supprimant l'usage des fonctions trigonométriques : la transformation en coordonnées sphériques. En coordonnées cartésiennes, un plan est caractérisé par un point p et une normale \vec{n} ; en coordonnées sphériques, par une direction Θ, Φ et la distance du plan à l'origine ρ . Cette transformation perd la notion de position du polygone à l'intérieur du plan-support (fig. 3.13), on ne peut donc pas déterminer si une surface est quasi-coplanaire à une autre mais seulement si leurs plans supports sont quasi-parallèles. Nous proposons d'utiliser l'espace dual formé par l'ensemble des Θ, Φ de l'espace sphérique (Annexe A), de créer des regroupements géographiques dans cet espace afin de trouver rapidement les ensembles des plans

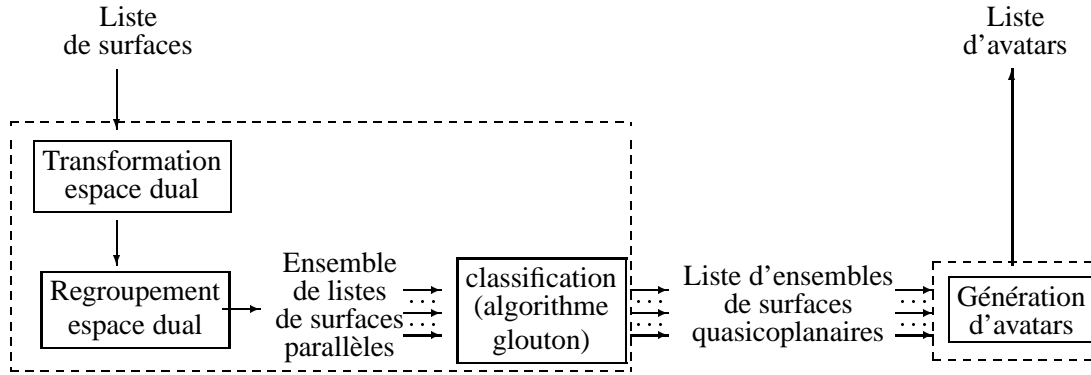


FIG. 3.12: Création d'avatar par l'utilisation d'un espace dual.

quasi-parallèles à une direction, puis de trouver les ensembles de surfaces quasi-coplanaires dans les ensembles précédents.

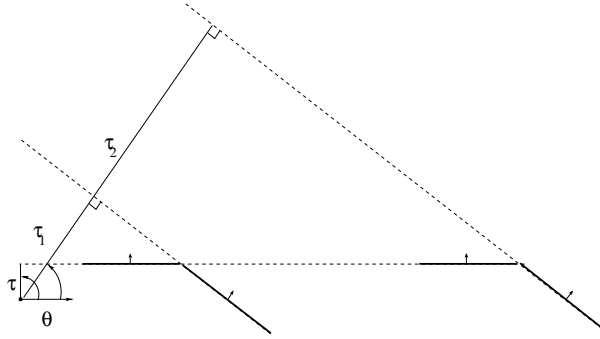


FIG. 3.13: Dépendance du paramètre de distance (ρ) à l'origine en coordonnées polaires.

Espace dual.

L'espace dual, tel que nous l'utilisons, est constitué des coordonnées Θ, Φ de l'espace sphérique (Annexe A), c'est-à-dire de la direction de la normale de chaque polygone. Cet espace à deux dimensions est cylindrique car l'axe Θ est de période 2π . De plus il possède deux singularités aux extrémités du tronc de cylindre en $\Phi = 0$ et $\Phi = \pi$. La distance entre un avatar et un plan est définie par l'angle entre les deux vecteurs normaux $\left(\vec{\mathcal{A}}_n, \vec{\mathcal{P}}_n \right)$. Dans l'espace dual, cette distance s'écrit :

$$\begin{aligned} \left(\vec{\mathcal{A}}_n, \vec{\mathcal{P}}_n \right) &= \arccos \left(\sin \Phi_1 \sin \Phi_2 \cos(\Theta_2 - \Theta_1) + \cos \Phi_1 \cos \Phi_2 \right) \\ &= \arccos \left(\frac{\cos(\Phi_2 - \Phi_1)(\cos(\Theta_2 - \Theta_1) + 1)}{2} + \frac{\cos(\Phi_2 + \Phi_1)(1 - \cos(\Theta_2 - \Theta_1))}{2} \right) \end{aligned}$$

Voisinage dans l'espace dual.

Afin de trouver rapidement les voisins d'une surface, nous avons utilisé une grille régulière, discrétisant dans l'espace dual les axes de longitude et de latitude de l'ensemble des directions. Chaque

cellule représente un ensemble de points proches. Par contre, la surface de chaque cellule n'est pas constante à cause de la déformation de la distance aux pôles. L'ensemble des points inférieurs à la distance α de (Θ, Φ) est borné par $\left[\frac{\Theta-\alpha}{\sin\Phi}, \Phi - \alpha\right] \times \left[\frac{\Theta+\alpha}{\sin\Phi}, \Phi + \alpha\right]$ avec $\Phi \in [0; \pi]$.

Un englobant encore plus large, au niveau de la structure de la grille régulière, est formé par l'ensemble des cellules du carré comprises entre les deux cellules correspondant aux bornes précédentes.

Regroupement dans l'espace dual.

Le regroupement de points dans l'espace dual est équivalent au regroupement de surfaces parallèles dans l'espace euclidien. Les groupes de points (fréquemment appelés *clusters*) ne sont pas uniformes sur l'ensemble de l'espace ; ceci est dû à la non uniformité de la distance définie. Une majorité des algorithmes classiques de regroupement se base sur ce pré-requis ; il faut donc faire attention à leur utilisation.

Il existe une vaste littérature à propos du regroupement pour l'optimisation d'accès à des bases de données importantes [CDG⁺89, The97, CLSS97]. Celle qui nous intéresse est le regroupement de points, sujet traité dans la segmentation d'images en couleurs [Cel90, iOKS80] Nous avons essayé plusieurs algorithmes classiques de regroupement de points dans un espace quelconque :

- **Les nuées dynamiques** : placement de germes de classes qui vont collecter progressivement les voisins proches et modifier itérativement les germes des classes. Les nuées dynamiques ne peuvent pas être utilisées car elles nécessitent la définition d'une distance intra-classe, d'une distance inter-classe et d'un nombre connu à l'avance de groupes (ou d'un nombre maximal de groupes). La première distance est proprement définie, la seconde pourrait être estimée mais le nombre de groupes n'est absolument pas connu avant sa génération. De plus, dans cet algorithme, tous les points doivent être regroupés, ce qui n'est pas notre cas ; il est difficile d'adapter l'algorithme afin de régler ce problème.
- **Analyse en composantes principales (ACP)** : transformation linéaire du problème afin de décorréler au maximum les axes de représentation des données. L'ACP n'est pas une technique de regroupement à proprement parler mais une façon de la simplifier. Notre espace n'étant pas uniforme, il n'est pas possible de trouver des axes de représentation linéaires, ce qui réduit de manière rédhibitoire son intérêt dans notre cas.
- **Diviser pour régner** : découpage d'un problème en plusieurs parties de façon hiérarchique, puis recombinaison aisée des traitements effectués sur chaque sous-partie. Il faut pouvoir trouver une méthode de découpage efficace, c'est pourquoi elle est fréquemment utilisée en association avec une analyse en composantes principales.

Nous allons utiliser une adaptation des nuées dynamiques en lui associant une légère relaxation sur un ensemble de groupes de taille croissante. Ainsi, la contrainte du nombre de groupes définie à l'avance disparaît tout en conservant une adaptation itérative.

Notre algorithme de regroupement construit un groupe pour chaque surface n'appartenant pas déjà à un groupe. Ce groupe, ne contenant alors qu'un seul point, va essayer d'agréger l'ensemble de ses voisins. Si un voisin appartient déjà à un groupe, il est placé dans le groupe le plus intéressant au sens de la distance. Le déplacement d'un point d'un groupe à un autre induit la réévaluation des points de l'ensemble initial, ceci parce que les groupes ont été modifiés et peuvent réagréger de nouveaux points en se déplaçant légèrement (algo. 3.4).

L'algorithme peut, théoriquement, entrer dans une phase oscillatoire où un point "sauterait" d'un groupe à un autre en modifiant suffisamment les groupes pour qu'ils veuillent le réagréger ensuite. En pratique ce cas n'est jamais apparu dans l'ensemble de nos tests, il convient toutefois de vérifier l'apparition de cycles. Le nombre de relaxations est souvent très faible (moins de 10), ce qui permet à cet algorithme d'être rapide.

Algorithme 3.4 Regroupement en surfaces quasi coplanaires grâce à l'espace dual**Regroupement Par Espace Dual** sur S S un ensemble de surfaces (cluster)

```

tant que  $change=vrai$  faire
   $change \leftarrow faux$ 
  pour toutes les surfaces  $S$  de  $S$  faire
    si (  $\exists C_1/S \in C_1$  ) alors
      Créer un groupe de points  $C$ 
      ajouter  $S$  à  $C$ 
       $Nouveau = vrai$ 
    fin si
    pour tous les voisins potentiels  $s$  de  $S$  faire
      si (  $\|Cs\| < \epsilon$  ) alors
        si (  $\exists C_1/s \in C_1$  ) alors
          si (  $\|C_1s\| < \|Cs\|$  ) alors
            supprimer  $s$  de  $C_1$ 
            ajouter  $s$  à  $C$ 
             $change=vrai$ 
          fin si
        sinon
          ajouter  $s$  à  $C$ 
        fin si
      fin si
    fin pour
  fin pour
  si (  $Nouveau = vrai$  ) alors
    ajouter  $C$  à  $\{C\}$ 
  fin si
fin tant que
retourner  $\{C\}$ 

```

où ϵ est un seuil de distance minimale défini par l'utilisateur et $\|Cs\|$ la distance d'un point à un groupe de point selon notre définition du voisinage dans l'espace dual.

Espace dual et espace mémoire.

L'utilisation de l'espace dual n'est pas une nécessité. Il est possible de créer une grille, régulière ou non, dans l'espace euclidien afin de connaître le voisinage. Il suffit dans ce cas de découper l'ensemble des vecteurs normaux de la sphère unité en un volume unitaire où chaque cellule représente l'intersection d'un petit cube avec la sphère. La notion de distance est alors simplifiée par un produit scalaire entre la normale à l'avatar et celle du polygone

$$\left(\overrightarrow{\mathcal{A}_n}, \overrightarrow{\mathcal{P}_n} \right) = \arccos \left(\overrightarrow{\mathcal{A}_n} \cdot \overrightarrow{\mathcal{P}_n} \right)$$

La distance peut, tout comme dans l'espace dual défini précédemment, être bornée. Les bornes d'angles sur la sphère unité sont réduites à la valeur de l'angle en lui-même sur les trois directions. L'ensemble des cellules dans lequel se trouvent tous les points inférieurs à une distance α d'un point (x, y, z) se situe dans la boîte $[x - \alpha, y - \alpha, z - \alpha] \times [x + \alpha, y + \alpha, z + \alpha]$. L'espace euclidien des directions discrétisé par une grille régulière est vide dans la plupart des cellules, car seule la surface de la

sphère unité est utile. L'espace mémoire perdu est une fonction cubique de la discrétisation utilisée, ce qui devient rédhibitoire dans certaines applications.

Classification en ensembles quasi-coplanaires.

Lorsque l'ensemble des surfaces quasi-parallèles est fixé, subdiviser cet ensemble en surfaces quasi-coplanaires repose uniquement sur l'opérateur de coplanarité (cf. §4.1, page 51). Les ensembles étant relativement réduits par rapport à la taille de la scène, il n'est pas utile d'optimiser fortement cette partie de l'algorithme.

Nous avons, dans notre réalisation informatique, utilisé le premier algorithme glouton de regroupement de surfaces sur les ensembles de surfaces quasi-parallèles.

5 Génération d'avatars à partir de surfaces coplanaires.

Nous avons découpé les algorithmes de génération d'avatars en deux grandes étapes : la découpe en ensembles de surfaces quasi-coplanaires et la génération d'avatars à partir de ces ensembles. Cette seconde étape peut être ramenée à la recherche de polygones englobants en deux dimensions. À nouveau, le nombre de résultats possibles est important et le jugement de la qualité d'un résultat doit dépendre de paramètres à définir. Le regroupement de surfaces en avatars est un problème similaire au regroupement de points dans l'espace dual ou de regroupement de surfaces dans des *clusters*.

Dans un premier temps, nous allons définir les paramètres qui régissent la génération d'avatars. Nous verrons que les algorithmes tels que les nuées dynamiques ou de type "diviser pour régner" ne peuvent pas être utilisés à cause des paramètres dépendants de la taille de l'avatar. Nous proposerons alors une méthode originale permettant de passer des minimums locaux dans la recherche du maximum de cette fonction complexe.

5.1 Paramètres.

Le but des avatars de surfaces est d'obtenir une répartition de l'énergie plus proche de la réalité en simplifiant la géométrie de polygones dégénérés et en limitant le nombre de calculs exécutés. Les paramètres doivent être en accord avec ce but, nous en avons identifié trois :

- Nombre de polygones : la simplification de la quantité de calculs dépend directement du nombre de polygones internes de l'avatar. Un seul calcul de facteur de forme et de visibilité sera factorisé pour l'ensemble des polygones internes.
- Régularité : Plus un polygone est régulier, plus le calcul de facteur de forme est précis. Mais cette précision est aussi dépendante de l'algorithme de calcul de facteur de forme, il est donc difficile de définir de façon générale l'impact de la régularité.
- Taux de remplissage : certains calculs inutiles risquent d'être exécutés dans des endroits vides de l'avatar. Le gain en temps de calcul dépend du rapport de la quantité de calculs utiles sur la quantité de calculs inutiles, le taux de remplissage permet d'estimer ce rapport.

La surface inutilisée d'un avatar dépend à la fois du taux de remplissage et de la surface de l'avatar. Plus cette surface est grande, plus il risque d'y avoir de calculs inutiles. Un avatar de taille très réduite a intérêt à factoriser un maximum de calculs car le gain en précision sera de toute manière négligeable. Nous avons donc choisi de faire dépendre les paramètres de la taille de l'avatar, ceci par une simple fonction linéaire bornée. Pour chaque paramètre, il est donc défini une valeur minimale, une valeur maximale et ainsi que l'aire des surfaces pour atteindre ces valeurs (fig. 3.14).

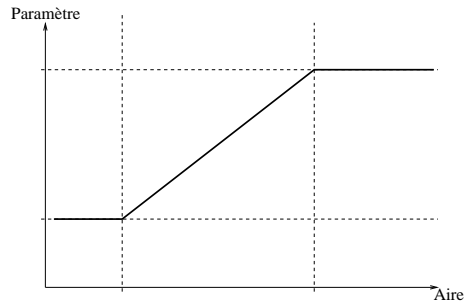


FIG. 3.14: Courbe d'interpolation linéaire bornée pour la valeur des paramètres.

5.2 Regroupements existants.

La recherche d'avatars dans un ensemble de surfaces quasi-coplanaires peut s'apparenter à la recherche d'un maximum dans une fonction très variable comportant une multitude de maximums locaux :

- toutes les surfaces n'appartiennent pas forcément à un avatar ;
- les avatars créés doivent répondre aux critères définis (cf. §5.1, page 57) ;
- le nombre d'avatars n'est pas fixe ;
- l'algorithme de génération doit se rapprocher de la (d'une) solution optimale décrite dans (cf. §3.2, page 50).

Les deux propriétés principales, maximisation du taux de remplissage et minimisation du nombre de polygones, dans la recherche de la solution étant antagonistes, les algorithmes habituellement utilisés ne peuvent fonctionner correctement dans notre cas :

- Descente de gradient : notre fonction possède trop de minimums locaux. Le critère de qualité d'une solution est vague, la définition du gradient l'est encore plus. Il faut une solution initiale que nous ne possédons pas.
- Nuées dynamiques : elles nécessitent d'avoir une idée approximative du nombre de groupes à créer. Pas de notion précise de distance inter-classes.
- Diviser pour régner : difficultés à gérer les groupes vides. Maximisation ou minimisation d'une fonction, mais pas les deux.
- Échantillonnage brutal : temps de calculs rédhibitoires. Dans notre cas, le nombre de degrés de libertés est trop élevé et il est difficile d'interpoler entre deux solutions.

5.3 Regroupement balancé.

Il nous faut un algorithme qui soit capable de minimiser une fonction et de maximiser une autre en même temps, tout en évitant de tomber dans les nombreux minimums et maximums locaux à chaque fonction. Nous avons développé un algorithme basé sur une relaxation pour se guider vers la (une) solution optimale, relaxation poussée par l'inertie pour essayer de sauter certains minimums/maximums locaux (algo. 3.5). Le principe de notre algorithme est de centrer l'avatar en un endroit pertinent puis d'agréger les polygones proches en essayant de sauter les minimums locaux jusqu'à la résolution des propriétés désirées.

Algorithme 3.5 Génération d'avatars à partir de surfaces quasi-coplanaires

```

Créer un Avatar  $\mathcal{A}$ 
Ajouter toutes les surfaces  $s$  de  $\{S\}$ 
PolygonesUtilisés  $\leftarrow \|\{S\}\|$ 
    Décime l'avatar en le centrant sur un point pertinent
tant que PolygonesUtilisés  $> 1$  et  $\mathcal{A}$  est valide faire
    | Calcule le nouveau centre de l'avatar avec PolygonesUtilisés polygones internes
    | Trie les polygones par la distance au centre
    | Réduire PolygonesUtilisés Décime le nombre de polygones de l'avatar
    | Calcule l'enveloppe convexe de l'avatar de PolygonesUtilisés polygones
fin tant que
    Réagrège les polygones proches de l'avatar
Calculer le centre de l'avatar
Trier les polygones de l'avatar par rapport à la distance au centre
tant que l'avatar est invalide et le nombre d'essais n'est pas maximal faire
    | Incréments PolygonesUtilisés Agréger les polygones les plus proches
    | Calculer l'enveloppe convexe de l'avatar
    | Vérifier que l'avatar est correct
    | si ( l'avatar n'est pas correct ) alors
    | | Incréments le compteur d'essais
    | fin si
fin tant que
si ( l'avatar est valide ) alors
    | Finaliser l'avatar avec le dernier nombre valide de polygones
sinon
    | Supprimer les polygones de la liste d'avatars
fin si

```

en définissant un avatar valide comme :

$$\mathcal{A}_r < \varepsilon_r(\mathcal{A}) \text{ et } \mathcal{A}_t < \varepsilon_t(\mathcal{A})$$

6 Résultats.

Cette section décrit l'utilisation d'exemples de scènes calculées avec et sans avatars de surfaces afin de quantifier les gains obtenus par cette technique.

6.1 Résultats visuels.

Dans l'exemple des engrenages déjà présenté dans la section "Approximation du facteur de formes" (cf. §1.2, page 42), on remarque l'intérêt de regrouper un ensemble de polygones coplanaires afin d'améliorer la qualité de la solution.

La première image est la solution originale calculée sans avatars de surfaces (fig. 3.15a) : on remarque les stries verticales dues à la géométrie dégénérée du modèle initial (fig. 3.2a). La génération automatique d'avatars permet de regrouper l'ensemble des surfaces coplanaires de chaque face de l'engrenage en deux avatars (fig. 3.15b). La troisième image en bas à gauche représente la géométrie de l'avatar frontal telle qu'elle est utilisée lors du calcul de la solution (fig. 3.15c).

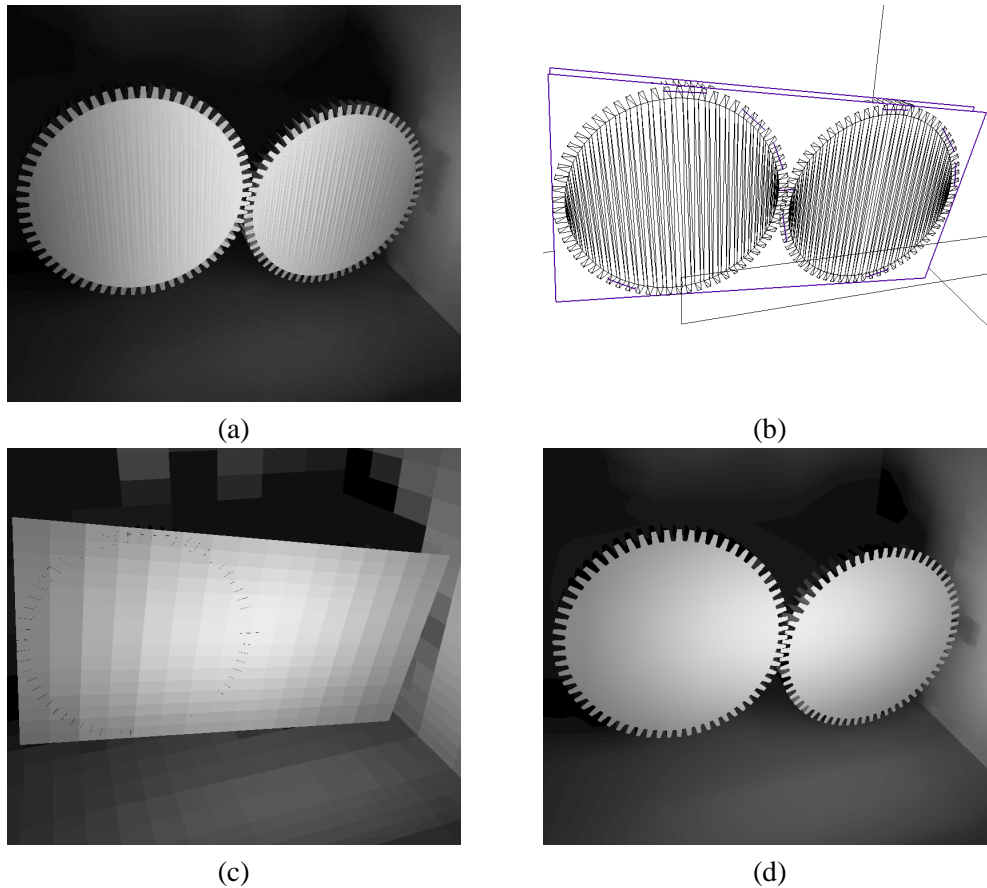


FIG. 3.15: Exemple d'amélioration visuelle induite par l'utilisation des avatars de surfaces.

Les artefacts noirs visibles (positionnés vaguement en forme de cercle) sont les petits avatars créés sur les dents de l'engrenage. Ils sont normaux et disparaissent en utilisant la visualisation basée sur le masquage par pochoirs (fig. 3.15d).

On peut remarquer visuellement les limitations des approximations des échanges énergétiques en affichant les niveaux hiérarchiques reliant les liens établis durant la solution (fig. 3.16). Le nombre de liens est beaucoup plus faible et les polygones utilisés dans l'échange sont plus homogènes dans le cas des avatars de surfaces. Chaque petit polygone dégénéré crée une erreur potentielle dans le transfert et la somme de toutes ces petites erreurs (si elles ne se compensent pas) peut être non négligeable dans le transfert global d'énergie.

Le deuxième exemple illustre l'avantage de regrouper des polygones limitrophes au sein d'un seul avatar. Dans l'image de la salle à manger sans avatars, on remarque d'importantes discontinuités aux jonctions des polygones triangulaires muraux, ceci malgré l'interpolation finale (fig. 3.17a). Ces discontinuités de jonctions disparaissent naturellement en utilisant les avatars car la répartition de l'énergie est plus uniforme sur l'ensemble des murs (fig. 3.17b).

6.2 Temps de création et de solution.

Les temps de calculs dépendent beaucoup du nombre d'avatars créés, du temps de création et du temps initial de la solution. Pour des scènes de faible complexité (centaine de polygones), l'intérêt

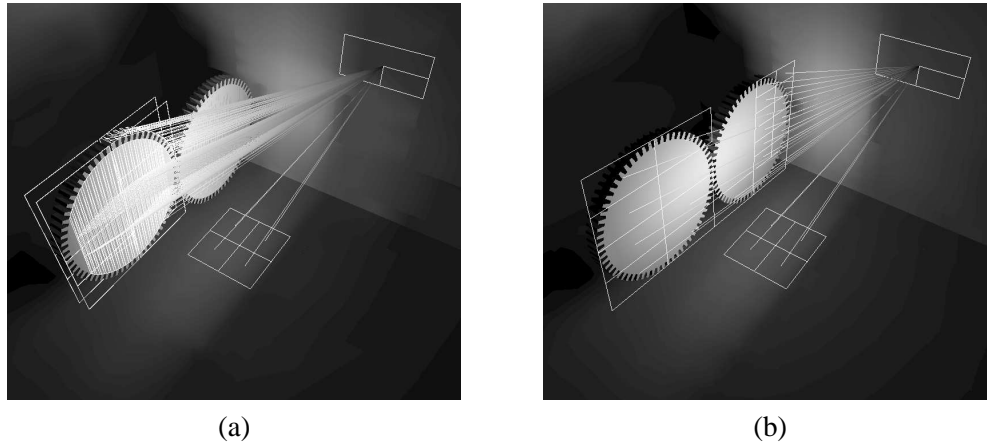


FIG. 3.16: Niveaux hiérarchiques d'établissement des liens. (a) Sans avatars. (b) Avec avatars.

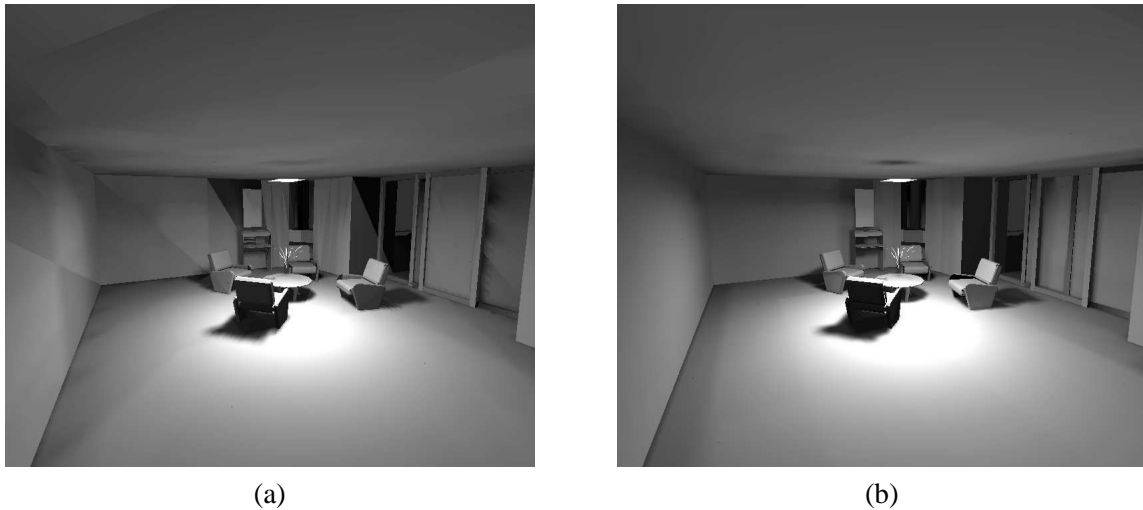


FIG. 3.17: Suppression de discontinuités entre polygones jointifs par l'utilisation des avatars de surfaces. (a) Solution sans avatars. (b) Solution avec avatars.

des avatars de surfaces est limité. Par contre, pour des scènes de complexité supérieure (au delà du millier de polygones), même un petit nombre d'avatars permet de réduire les temps de calculs.

Nous avons utilisé une scène avec un mélange d'éclairage direct et indirect, un nombre moyen de polygones mais comprenant des éléments à des échelles variées. La création d'avatars permet de regrouper les petits polygones d'une manière plus précise que les regroupements de surfaces.

Le temps de création moyen des avatars est relativement faible par rapport au temps de calcul de la solution est permet de gagner de 20 à 80% sur le temps de calcul de la solution. Lorsque le paramètre de taux de remplissage est relativement étendu, les temps de création des avatars de surfaces est plus faible que dans le cas d'un paramètre élevé mais moins étendu. Le nombre d'avatars est alors plus élevé (~ 1000 contre 150) mais ne modifie quasiment pas le temps de calcul de la solution.

La différence de qualité visuelle est aussi imperceptible, toujours améliorée par rapport à la solution initiale comme décrit dans la section précédente.

Pour un temps de création plus élevé il n'y a pas de gain supplémentaire de qualité ni de réduction de temps de calculs. Ceci est d'autant plus vrai que les scènes sont complexes. Par contre, lorsqu'il y a un nombre d'avatars important, le nombre de liens nécessaire à la solution est plus élevé. Ce qui utilise des ressources mémoire inutilement par rapport à un nombre d'avatars plus faible et mieux choisis (tab. 3.3).

Il y a donc un compromis à faire entre temps de création et nombre d'avatars créés, étude de notre prochaine section.

Simulation sans avatars :

<u>Génération d'avatars</u>	<u>Calcul de la solution</u>		
	itération	Nombre de liens	temps (s.)
0	1	15017	305.1
	2	910256	3560.6
	3	1254132	2546.4

TAB. 3.1: Calcul d'une solution de radiosité sans avatars

Première simulation avec avatars :

<u>Génération d'avatars</u>		<u>Calcul de la solution</u>		
Critère de taux de remplissage	[0.9 – 1.0]	itération	Nombre de liens	temps (s.)
Temps de création	80 s.	1	12763 (-15%)	30.7 (-89%)
Nb d'avatars	151	2	756086 (-17%)	2845.5 (-20%)
Nb de polygones des avatars	790	3	1066055 (-15%)	1986.4 (-22%)

TAB. 3.2: Calcul d'une simulation de radiosité avec avatars de surfaces paramétrée par un taux de remplissage élevé.

Seconde simulation avec avatars :

<u>Génération d'avatars</u>		<u>Calcul de la solution</u>		
Critère de taux de remplissage	[0.2 – 1.0]	itération	Nombre de liens	temps (s.)
Temps de création	61.9 s.	1	15267 (-1.7%)	35.8 (-88%)
Nb d'avatars	958	2	760446 (-16%)	2649.4 (-25%)
Nb de polygones des avatars	4436	3	1118670 (-11%)	1952.5 (-23%)

TAB. 3.3: Calcul d'une simulation de radiosité avec avatars de surfaces paramétrée par un taux de remplissage étendu.

6.3 Paramètres.

Nous avons testé l'impact du paramètre de taux de remplissage minimal accepté dans la génération automatique d'avatars. La technique de regroupement de surfaces quasi-coplanaires utilisée est le regroupement par espace dual qui donne les meilleurs résultats.

Pour étudier les différents paramètres, nous avons utilisé trois scènes ayant des caractéristiques différentes. Les sections suivantes décrivent les scènes utilisées, ainsi que les hypothèses sur l'algorithme de génération. La discussion des vastes résultats suivra leur présentation.

La Rue.

La scène RUE décrit une rue complète avec les façades d'immeubles, quelques voitures stationnées, une arche et des luminaires. Elle est composée de 40573 polygones initiaux, relativement petits, ordonnés en surfaces (trottoirs, murs, façades, sols, ...) tels que cette scène doit se prêter facilement à la génération d'avatars (fig. 3.18).

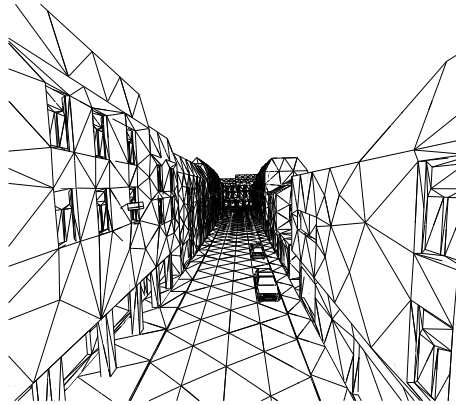


FIG. 3.18: Scène de rue comportant beaucoup de surfaces coplanaires.

Albisia.

La scène de l'ALBISIA est une modélisation de l'arbre avec 47326 petits polygones. Chaque feuille comporte 6 polygones fortement coplanaires et chaque branche est composée de plusieurs feuilles faiblement coplanaires (fig. 3.19). Ce type de modèle géométrique très fourni, avec une faible cohérence entre les différents éléments est un cas extrême pour notre algorithme qui doit essayer de retrouver des plans dans une "soupe" de polygones.

Nous avons testé cette scène à deux échelles différentes (fig. 3.20) afin de voir l'influence de la taille dans l'algorithme de génération d'avatar proposé. La scène de petite taille est identique à la grande, seul l'arbre a été supprimé de la pièce de gauche, réduit en bonzaï et posé sur l'étagère de la pièce de droite (fig. 3.20b).

Régularité.

La régularité des avatars de surface (cf. §2.1, page 44) n'est pas directement liée à l'algorithme de construction. Elle est logiquement entièrement décorrélée des seuils fixés. L'étude des courbes de régularité, en fonction du paramètre variable utilisé dans notre algorithme de génération des avatars (le taux de remplissage), montre effectivement cette absence de corrélation (fig. 3.21).



FIG. 3.19: Arbre dont les feuilles forment un ensemble de surfaces faiblement coplanaires.

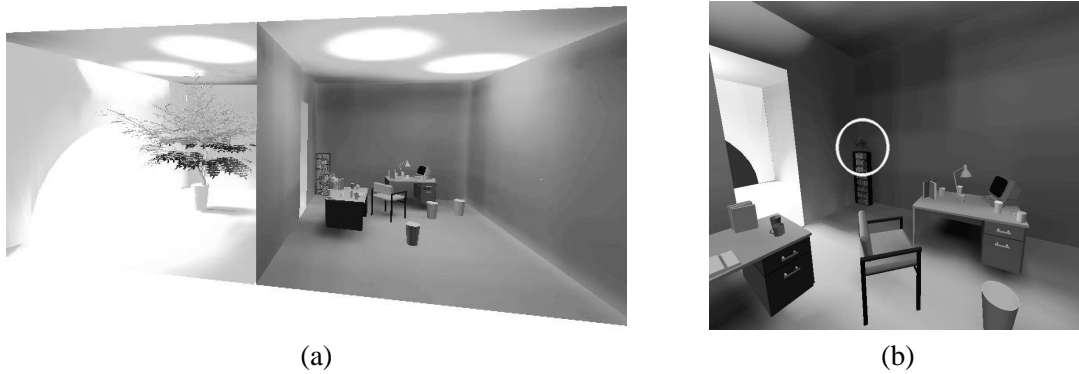


FIG. 3.20: Scènes ALBISIA comportant un arbre à deux échelles différentes. (a) Taille importante (de l'ordre du mètre). (b) Taille réduite (de l'ordre du centimètre).

La courbe correspondant à la scène de la RUE est très crénelée alors que celles correspondantes à la scène de l'ALBISIA ont une forme moins discontinue. Même si deux de ces courbes semblent avoir de faibles variations, le point important est de remarquer la non-similitude entre les courbes. Ces différences montrent effectivement qu'il n'est guère possible de trouver une expression de la régularité en fonction des avatars générés.

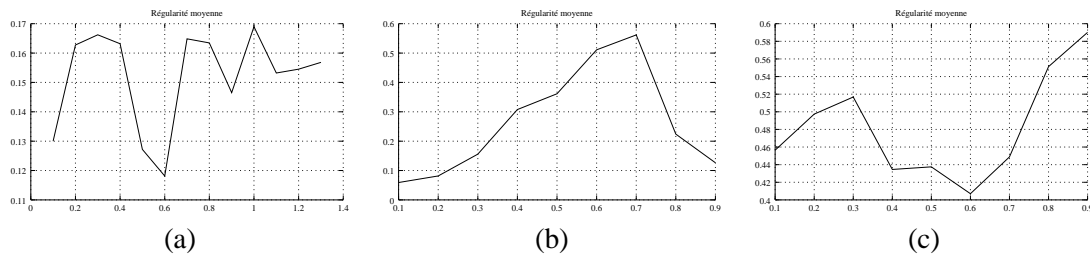


FIG. 3.21: Non corrélation entre régularité et taux de remplissage de l'avatar sur trois différentes scènes. (a) RUE. (b & c) ALBISIA petite et grande taille.

Algorithme utilisé.

La fonction de régularité n'étant pas liée aux paramètres de notre algorithme de génération d'avatars (cf. §6.3, page 63), nous n'avons pas mis de bornes sur la régularité des avatars créés afin de ne pas perturber le comportement des autres paramètres étudiés.

Nous n'avons pas non plus fixé de nombre de polygones minimal par avatar (soit deux au minimum), mais les statistiques de génération sont très dépendantes de ce seuil au niveau des résultats de gain de temps et de qualité.

Il est à noter que les paramètres de regroupement en surfaces coplanaires ont un rôle non négligeable dans les résultats obtenus ; nous les avons fixés aux valeurs moyennes que nous utilisons habituellement.

Courbes des paramètres étudiés.

Nous avons choisi 6 paramètres afin de juger la qualité de la génération des avatars : le nombre d'avatars généré, le nombre de polygones total inclus dans un avatar, le nombre de polygones moyen par avatar, l'aire moyenne des avatars, le taux de remplissage moyen effectivement créé après génération. Même si les temps de calculs ont déjà été discutés précédemment. Nous les avons inclus car la forme de leurs courbes est intéressante.

Nous n'avons pas utilisé de critère de différences en qualité visuelle car la génération est un processus en amont du calcul de la solution. Ces différences dépendent fortement de la géométrie de la scène mais aussi des approximations que l'on utilise dans le calcul des facteurs de formes. L'hypothèse d'utiliser un calcul particulier serait trop forte par rapport aux résultats obtenus.

Analyse.

Les courbes décrivent, en règle générale, le comportement que l'on pouvait attendre de l'impact du taux de remplissage dans l'algorithme proposé de génération d'avatars (figs. 3.22, 3.23, 3.24). Le nombre d'avatars croît puis décroît en forme de cloche. Au début, le taux de remplissage est faible : il est donc possible de faire peu d'avatars de surface importante avec beaucoup de polygones. Ensuite, plus le seuil de taux de remplissage augmente, plus les grands avatars sont découpés en avatars de taille plus réduite en perdant quelques polygones isolés. Ceci augmente le nombre d'avatars mais réduit le nombre de polygones utilisés. Finalement, lorsque le taux de remplissage devient très contraignant, beaucoup d'avatars ne peuvent plus exister, diminuant le nombre d'avatars ainsi que le nombre de polygones utilisés.

Les avatars de taille importante sont ceux qui ont le plus de probabilités d'avoir un taux de remplissage faible. Ce sont donc majoritairement eux qui disparaissent, réduisant par la même occasion la surface moyenne des avatars.

Dans la scène ALBISIA de grande taille, on remarque une saute des courbes vers le taux de remplissage égal à 0.7, ceci est dû au grand nombre d'avatars comportant très peu de polygones (2-3) atteint rapidement. Ces avatars disparaissent en même temps dû à l'uniformité du modèle, laissant moins d'avatars d'une surface moyenne beaucoup plus élevée.

Les temps de calculs croissent rapidement en fonction du taux de remplissage désiré, ceci de façon quasi linéaire, sigmoïdale et exponentielle pour les trois scènes testées. Il n'est donc pas possible de tirer une conclusion plus précise que "plus le taux de remplissage désiré est élevé, plus le temps de calcul va être important". Les échelles de valeurs sont relativement différentes, L'algorithme prenant beaucoup plus de temps pour des objets de petite taille car les cellules de proximité contiennent beaucoup plus de polygones.

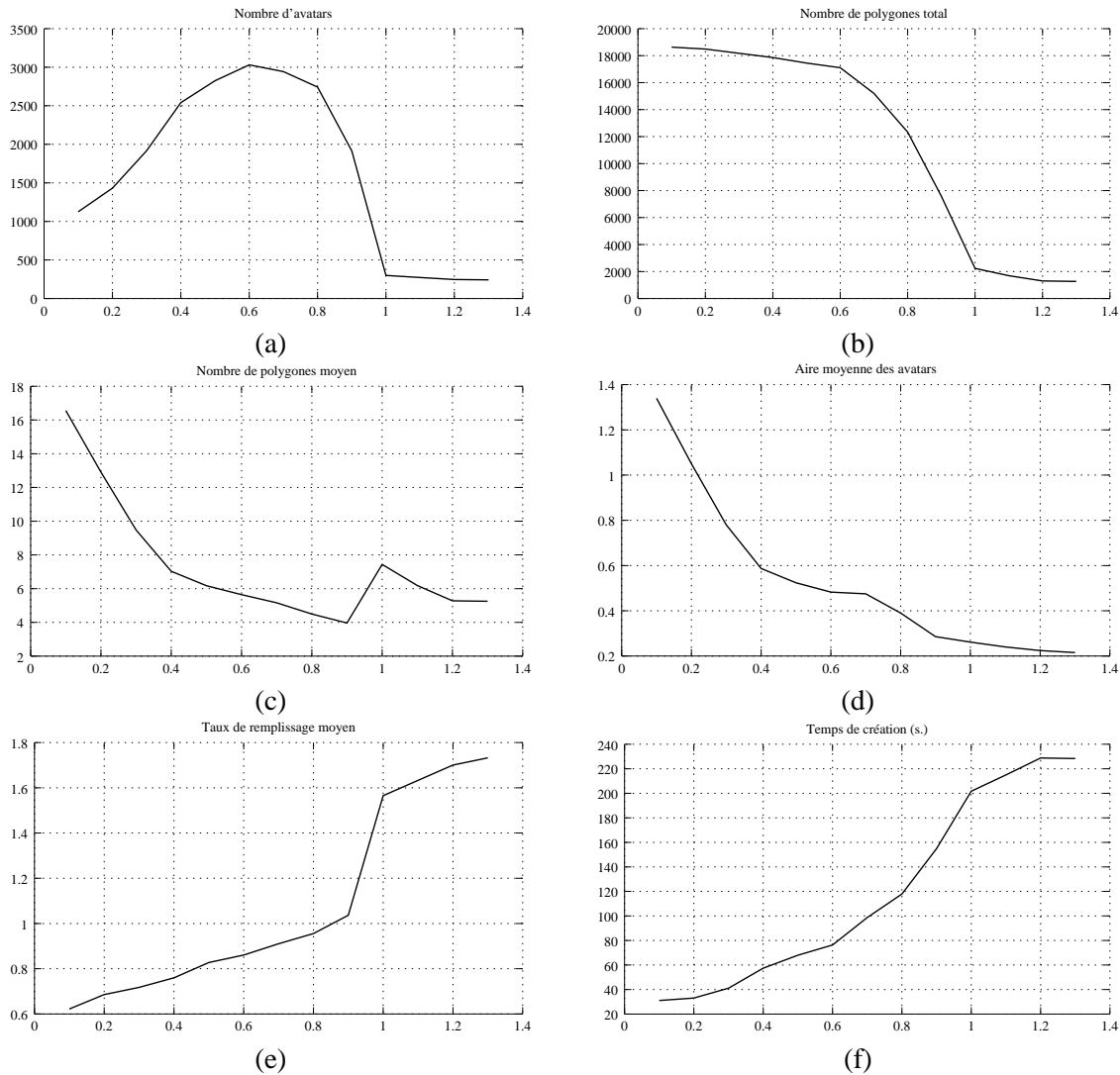


FIG. 3.22: Variation de la génération d'avatars en fonction du paramètre de taux de remplissage pour la scène RUE

Position optimale.

On remarque donc qu'il y a une position favorable où le nombre d'avatars est élevé, les temps de calculs encore loin de leur maximum et un taux de remplissage effectif correct. Cet "optimum", même s'il est légèrement décalé pour chacune des scènes, reste très proche de la valeur médiane 0,5. Les caractéristiques citées restent valables pour toutes les courbes à cette valeur, même si l'aire des avatars est faible, ce qui a tendance à créer de nombreux petits avatars.

Nous préconisons donc l'utilisation de la valeur médiane pour toutes les scènes, en lui adjoignant un seuil sur la régularité. Effectivement, on peut remarquer sur les courbes correspondantes de régularité (fig. 3.21) que la valeur de régularité n'est pas forcément la meilleure à cette valeur.

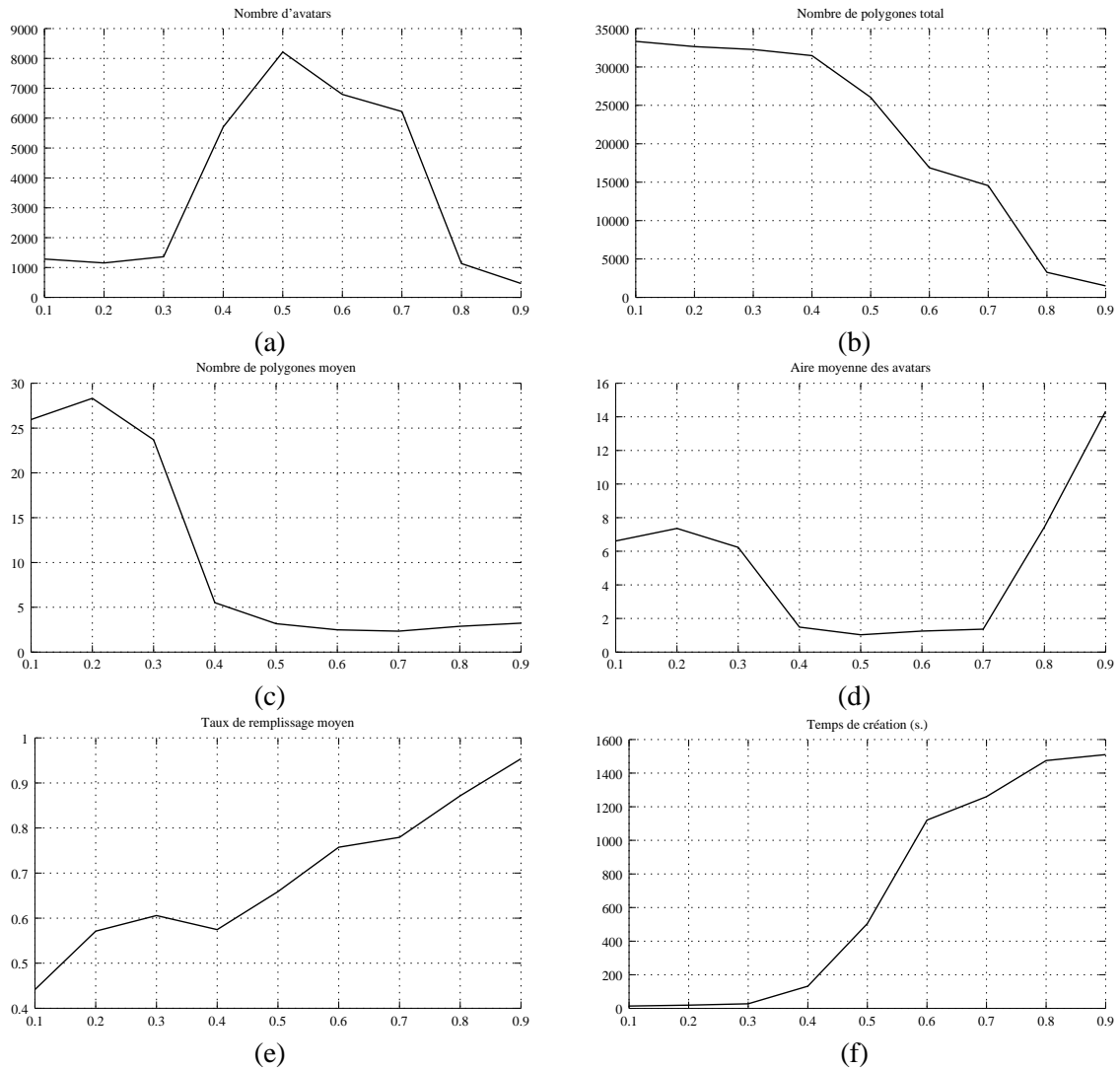
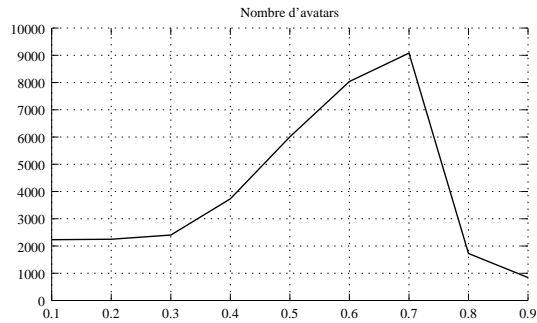


FIG. 3.23: Variation de la génération d'avatars en fonction du paramètre de taux de remplissage pour la scène ALBISIA (petite taille)

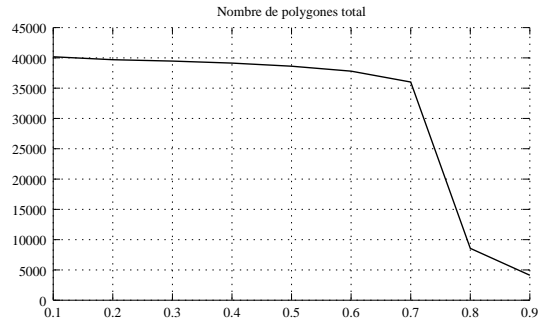
7 Conclusion.

La fonction de radiosit  souffre d'un nombre important d'approximations durant son  valuation,   la fois   cause de la g om trie de la sc ne et du calcul m me du facteur de forme. Les avatars de surfaces permettent de r duire la taille de la description g om trique de la sc ne en contr lant la qualit  du maillage g n r  et ainsi de limiter les approximations du facteur de forme. Tout en simplifiant la complexit  de la sc ne, donc des temps de calculs, les avatars de surfaces permettent d'am liorer la qualit  de la solution.

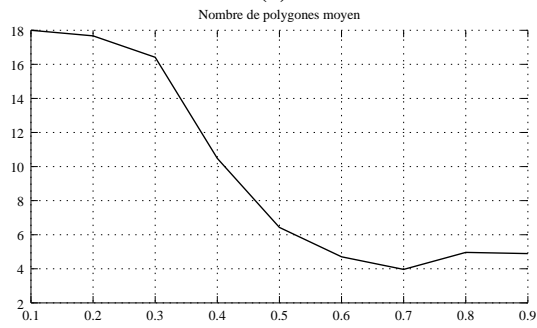
Nous avons pr sent  diff rentes m thodes de g n ration d'avatars de surfaces plans, ainsi qu'une  tude des diff rents param tres qui r gissent ces algorithmes. Une solution favorable pour une majorit  des sc nes permet de simplifier le contr le de la g n ration.



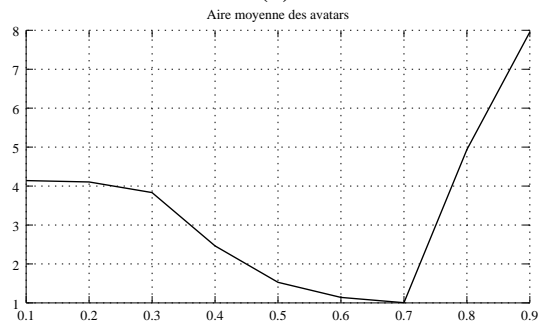
(a)



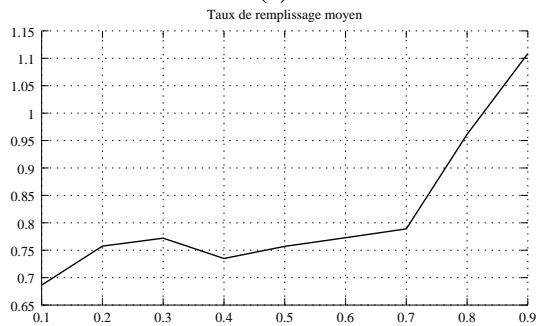
(b)



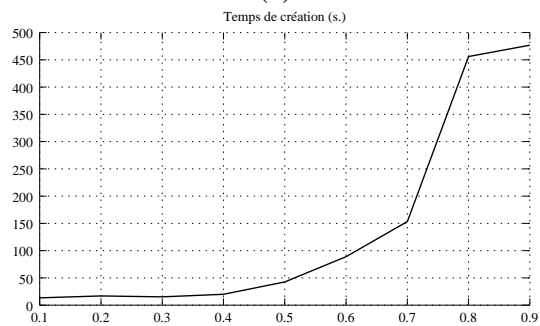
(c)



(d)



(e)



(f)

FIG. 3.24: Variation de la génération d'avatars en fonction du paramètre de taux de remplissage pour la scène ALBISIA (grande taille)

Le raffinement

*L*e raffinement est en radiosit  hierarchique est un processus de subdivision   la fois g om trique et  nerg tique. Il permet d'obtenir une estimation de la solution de radiosit  suffisamment pr cise en utilisant le minimum de ressources possibles. La majorit  des approximations faites repose sur le calcul du facteur de forme entre deux  l ments, qui caract rise leur  change radiatif. Lorsqu'il int gre la visibilit , le raffinement est le composant central de l'algorithme de radiosit  hierarchique car il d termine la pr cision de la solution ainsi que les temps de calculs.

Par effet de bord, le raffinement est  quivalent   la subdivision des  l ments hierarchiques afin d'arriver   des estimations de facteurs de formes et de facteurs de visibilit  suffisamment pr cis. La solution obtenue est une fonction d finie par morceaux. Chaque morceau ( l ment hierarchique) poss de une repr sentation locale de la fonction de radiosit . L'*oracle de raffinement* est l'heuristique du raffinement qui d cide si cette repr sentation locale par  l ment hierarchique est suffisamment pr cise ou non.

le raffinement reste un sujet de recherche important   cause de la grande vari t  de configurations g om triques et radiom triques impliqu es dans l'ensemble des  changes  nerg tiques test s. De plus, malgr  les nombreuses recherches d j  effectu es sur ce sujet, il n'existe pas un oracle de raffinement universel qui fonctionne correctement pour l'ensemble des sc nes.

L'imbrication des diff rentes raisons possibles de raffinement rend l' tude du probl me complexe. Nous pr sentons dans ce chapitre une structuration du raffinement sous forme d'un ensemble de t ches  l mentaires trait es par des *outils*. La d composition de ce probl me permet de le formaliser en ayant une approche plus syst matique. Nous proposons alors une structure de recomposition sous

forme d'un graphe permettant une modularité très utile dans la recherche d'un algorithme performant. Nous listerons un ensemble des outils les plus utilisés et nous terminerons enfin par un exemple d'oracle de raffinement.

1 Le raffinement progressif.

La radiosité hiérarchique permet le contrôle de la précision de la solution calculée. Si l'erreur entre la solution calculée et la solution réelle est jugée trop importante, le raffinement va permettre de la réduire. La difficulté réside dans la décision de raffiner ou non. La connaissance de la différence entre ces deux solutions permettrait de résoudre ce problème trivialement. Hélas, cette erreur n'a pas d'expression simple.

L'oracle décide à quel niveau hiérarchique les éléments de la scène peuvent établir leurs échanges radiatifs, ces échanges sont symbolisés par des *liens*. Malgré la simplicité apparente du choix du raffineur : "le lien estime-t-il correctement l'échange radiatif à ce niveau hiérarchique ou doit-il être subdivisé ?", les oracles existants ne sont utilisables que dans des configurations spécifiques et sont d'approche souvent difficile. C'est une des principale raisons du lent développement de l'algorithme comparativement aux autres algorithmes de simulation d'éclairage global.

Un premier ensemble de critères suppose que l'erreur commise sur un élément hiérarchique est proportionnelle à l'irradiance qu'il reçoit. Le plus simple critère est un seuil sur cette irradiance (*critère BF*) [HS90]. Il a l'avantage d'être rapide et facile à utiliser, par contre il n'est pas stable par rapport à un changement d'échelle de la gamme d'intensité utilisée car la valeur du seuil est absolue. De plus, l'intégration de la visibilité n'est pas décrite de façon explicite dans la méthode.

Une deuxième catégorie de critères consiste à trouver une borne ou un encadrement de l'erreur commise [HS98, SAS92]. Ces calculs, souvent coûteux, tendent à trop raffiner car les bornes d'erreur s'éloignent rapidement. Les encadrements trouvés restent trop éloignés de la solution réelle, essentiellement à cause de la visibilité. La troisième catégorie estime l'impact du raffinement sur la totalité de la solution [HSD94, LSG94], le coût en temps étant aussi très important.

1.1 Raisons de subdivision.

Un critère de raffinement est composé de différents facteurs de natures différentes. Typiquement la subdivision doit être plus importante le long des limites d'ombre et dans des zones où la variation de radiosité est importante. Ces deux facteurs sont presque indépendants et peuvent être traités séparément. Le nombre de facteurs semble assez faible, voici ceux que nous avons jugé les plus discriminants :

- **Visibilité** : les plus forts gradients d'une solution de radiosité se situent fréquemment au niveau des ombres résultant d'un ou plusieurs obstacles.
- **Énergie** : dans un contexte de visibilité totale, la répartition de l'énergie sur un élément de hiérarchie dépend de la distribution de l'émission de la source et de la géométrie relative des deux éléments.
- **Géométrie** : certaines configurations géométriques rendent les estimations d'échanges difficiles. Par exemple, l'interpénétration d'éléments hiérarchiques rend les estimations de calcul de facteur de forme difficiles.

1.2 Oracle de raffinement.

L'oracle de raffinement est l'heuristique qui décide si la fonction de l'énergie transférée de l'émetteur sur le récepteur est suffisamment proche de la base de fonctions utilisée. Selon sa décision il va

conserver le lien courant à ce niveau hiérarchique ou subdiviser un des éléments (ou les deux) pour tenter d'établir des liens à un niveau inférieur de manière plus précise.

L'oracle peut utiliser plusieurs critères de raffinement en fonction de la configuration géométrique des éléments ou de leurs caractéristiques spécifiques, il peut se représenter sous forme d'un sélecteur de critères. Les critères utilisés et l'ordre dans lesquels ils vont être employés sont dépendants de l'application et des résultats à obtenir. Un oracle de simulation d'éclairage destinée à un jeu vidéo cherchera une image de grande qualité visuelle et une solution énergétique relativement approximative, là où un calcul scientifique de ce même éclairage imposera une erreur énergétique minimale sans se soucier de la qualité visuelle des ombres.

Pour prendre en compte l'ensemble des critères possibles au sein d'un oracle, nous proposons de représenter le processus de raffinement comme une "discussion" entre l'oracle et une collection d'outils. L'oracle pose une question à un outil, et prend une décision en fonction de sa réponse (fig. 4.1). Les décisions possibles sont :

- poser une question à un nouvel outil ;
- établir un lien à ce niveau hiérarchique ;
- subdiviser l'un, l'autre ou les deux éléments hiérarchiques.

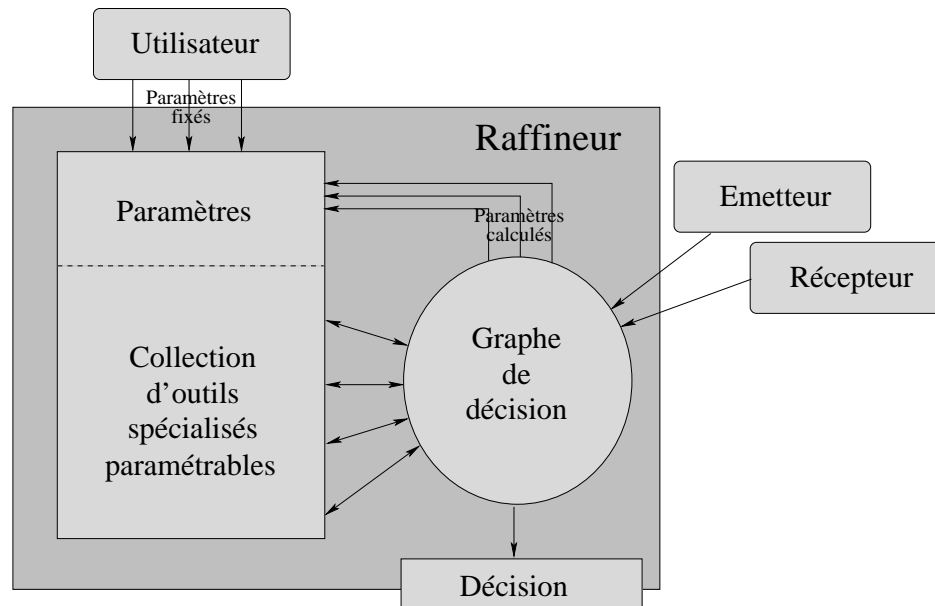


FIG. 4.1: Structuration du processus de raffinement en discussion entre un ensemble d'outils et l'oracle de raffinement.

Pour que l'utilisateur puisse influencer la discussion dans une direction qu'il privilégie, il peut modifier les paramètres des outils. Ces paramètres sont le média de communication entre le processus de raffinement et l'utilisateur.

2 Outils de raffinement.

Les outils sont les briques essentielles à la construction du processus de raffinement, ils possèdent tous la même structure générale et doivent répondre à une question simple ou exécuter une action basique. Il existe deux types de réponses possibles :

- **Les réponses précises** : ce sont les réponses attendues à la question après la réussite de l'exécution de l'outil. Par exemple, pour un outil qui demande une classification de la visibilité (cf. §1, page 98), les réponses peuvent être "la visibilité est totale entre l'émetteur et le récepteur" ou "la visibilité est partielle"... Les réponses précises sont crédibles à 100%.
- **Les réponses approximatives** : c'est l'ensemble des réponses qui indiquent qu'une réponse précise n'a pas pu être trouvée. Par exemple un outil peut répondre "je ne sais pas faire du tout", "c'est trop compliqué" ou encore "ca va prendre trop de temps"...

2.1 Outils d'interrogation.

Les outils d'interrogation sont les outils qui répondent à une question, ils forment le corps de la discussion et permettent à l'oracle de prendre sa décision. Nous avons distingué trois types d'outils d'interrogation correspondants aux différentes classes de problèmes impliqués dans le raffinement soit l'énergie, la visibilité et la géométrie (cf. §1.1, page 70).

Outils énergétiques.

Les outils énergétiques répondent à toute question concernant l'échange radiatif du lien en cours de raffinement. Ils peuvent par exemple déterminer si l'énergie du lien est suffisante, si l'apport d'énergie est visuellement discriminant, etc... La figure (fig. 4.2) montre un exemple de répartition de la fonction de radiosité non uniforme sur le récepteur du à la disposition des éléments et de leur énergie, c'est un cas que des outils énergétiques spécialisés peuvent tenter de déterminer.

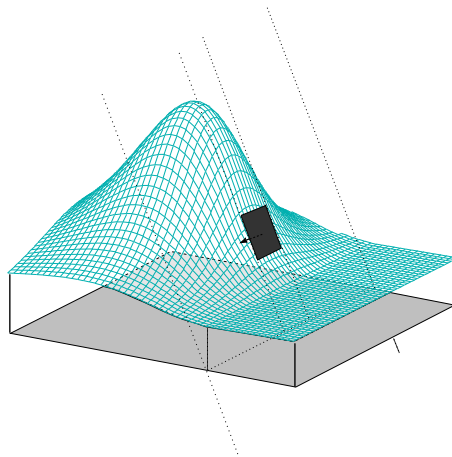


FIG. 4.2: Exemple de cas de répartition de l'énergie non uniforme à déterminer par un outil énergétique.

Dans cet exemple, il est possible de créer une multitude d'outils pour résoudre ce problème :

- **Échantillonnage** : prendre un certain nombre d'échantillons sur le récepteur et/ou l'émetteur et calculer pour chaque paire d'échantillons un échange énergétique point à point. En fonction de la distribution des valeurs énergétiques répondre à la question par "la répartition est homogène" ou "la répartition est hétérogène".
- **Disposition relative** : comparer l'échelle de l'énergie reçue au niveau du récepteur à la taille du récepteur. Une source très étendue aura une répartition énergétique uniforme sur un récepteur de taille inférieure.

- Calcul du gradient : calculer, estimer ou borner le gradient de la fonction de radiosité sur le récepteur de façon à vérifier qu'il ne dépasse pas une certaine valeur.
- ...

Outils de visibilité.

Les outils de visibilité doivent permettre de caractériser la visibilité entre deux éléments. Ils peuvent avoir deux actions, combinées ou dissociées :

- **Classification de la visibilité** : déterminer sans calculer de valeur numérique quelle est la classe de la configuration des deux éléments. Par exemple déterminer si la visibilité est totale, partielle ou nulle.
- **Évaluation de la visibilité** : estimer le facteur de visibilité entre les deux éléments, c'est à dire la proportion d'émetteur visible depuis le récepteur.

Les classifications seront décrites en détail dans le chapitre 5 (page 97).

Outils géométriques.

Les outils géométriques et les outils énergétiques sont parfois difficilement dissociables, car l'évaluation de l'échange radiatif dépend de la géométrie. Toutefois les outils géométriques doivent résoudre les problèmes qui n'utilisent pas la notion de flux d'énergie entre l'émetteur et le récepteur. Ce sont par exemple des problèmes de surface auto-intersectantes, de cônes de normales, etc...

Ils sont très utiles pour garantir l'utilisation des autres outils dans des conditions particulières. Par exemple, le calcul de facteur de forme entre deux surfaces auto-intersectantes peut amener des approximations très fortes selon la méthode de calcul utilisée, il convient donc d'appeler des outils de calcul de facteur de forme spécialisés dans le cas ou un objet géométrique de discrimination d'auto-intersection à répondu par la positive.

D'autres outils rentrent dans le cadre des outils géométriques, tels que : l'élimination des faces arrières (*backface culling*), la discrimination d'auto-intersection, le calcul de critères portant sur les angles solides ou la distance relative entre l'émetteur et le récepteur, etc...

2.2 Outils d'action.

Les outils d'action sont des outils particuliers qui exécutent une action, par exemple, subdiviser l'élément hiérarchique récepteur du lien en cours de raffinement. Il existe deux catégories d'outil d'action :

Outils de décision.

Ce sont les réponses possibles que l'oracle de décision peut prendre, ces outils doivent toujours se situer à la fin de la discussion.

- **Établissement de lien** : c'est la décision prise lorsque l'oracle estime que la représentation de la radiosité de l'émetteur sur le récepteur est suffisamment uniforme.
- **Subdivision** : lorsque la répartition de l'énergie n'est pas uniforme, il convient de subdiviser un des éléments hiérarchique pour diminuer l'erreur totale de la scène et s'approcher de la solution idéale. Le choix de l'élément à subdiviser (l'émetteur ou le récepteur) est aussi une des décisions de l'oracle de raffinement.
- **Destruction de lien** : un lien peut être détruit si l'oracle juge qu'il n'est plus pertinent dans la solution de radiosité.

Outils de commande.

Les outils de commande sont des ordres que l'oracle peut imposer au cours de la discussion, ils peuvent servir pour marquer un élément, pour imprimer des informations sur le cours de la discussion, etc... Leurs réponses sont succinctes, la réponse principale étant TRAVAIL EFFECTUÉ.

Ce sont essentiellement des outils de débogage, de visualisation, d'optimisation ou de calculs inconditionnels. Par exemple un outil très pratique est l'outil de marquage, les éléments hiérarchiques extrémités du lien en traitement sont marqués, ce qui permet de visualiser l'ensemble des éléments qui ont été traités en répondant à une série de questions.

Un autre outil de commande très utilisé est le calcul du facteur de visibilité. Ce n'est pas un outil de visibilité car il ne répond à aucune question, il exécute simplement le calcul.

3 Structure arborescente de l'oracle de raffinement.

L'oracle de raffinement pose des questions aux outils. En fonction des réponses qu'il obtient, il pose de nouvelles questions pour aboutir finalement à une décision. L'ensemble des questions posées et la décision forment une discussion. Nous proposons de représenter l'ensemble des discussions possibles par une structure de graphe où chaque noeud est une question, chaque arc une réponse et chaque feuille une décision (fig. 4.3). Nous appelons ce graphe représentant l'oracle de raffinement *graphe de décision*.

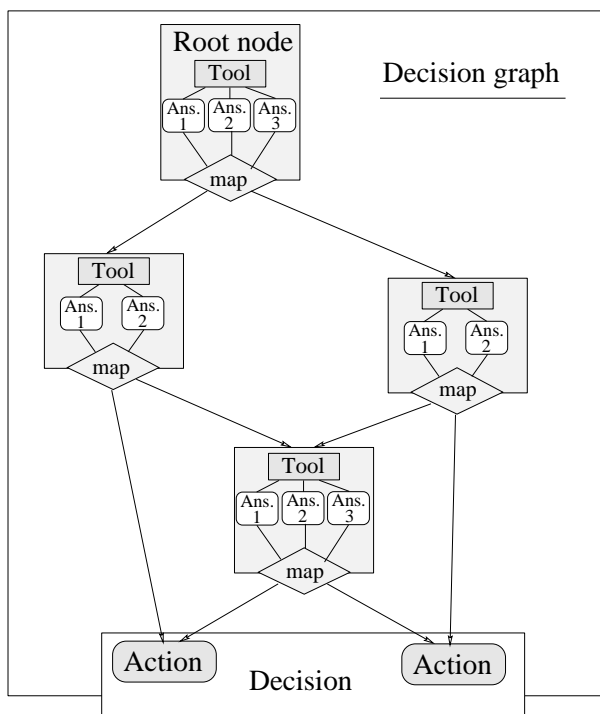


FIG. 4.3: Structure arborescente du graphe de décision.

Le graphe ne doit pas comporter de cycles car ceux-ci pourraient engendrer un processus de raffinement infini. Le graphe est donc un graphe directionnel sans cycle, nommé plus couramment sous le terme anglophone de *Directed Acyclic Graph (DAG)*.

3.1 Gestionnaire de données.

Au cours d'une discussion il est fréquent que différentes questions aient besoin de données communes. Par exemple, l'axe principal entre le récepteur et l'émetteur est utilisé pour des outils géométriques (orientation relative des éléments), énergétiques (estimation du facteur de forme) ou de visibilité (direction principale de répartition de rayons de visibilité). Or la modularité des outils empêche la discussion entre eux. Chaque outil devant pouvoir répondre à une question ou effectuer une action, sans à-priori sur l'environnement qui l'entoure.

Puisque le partage de données n'est pas naturel par la structure de graphe utilisée mais est nécessaire pour éviter les redondances, nous recourons à un gestionnaire de données qui est un média entre les outils. Ce gestionnaire permet de stocker les résultats intermédiaires utiles à certains outils. A chaque fois qu'un outil nécessite le calcul d'une donnée susceptible d'être utile à plusieurs outils, il demande au gestionnaire s'il la possède afin de ne pas la recalculer. Si le gestionnaire ne possède pas encore cette valeur, l'outil doit savoir la calculer et doit dire au gestionnaire qu'il vient de la calculer en lui donnant le résultat de son calcul.

Certaines données ne peuvent pas être jugées sur un critère uniquement binaire – déjà calculée ou non – mais sont une estimation plus ou moins forte de ce qu'elles représentent. Par exemple, le coefficient de visibilité est quasiment toujours une approximation de la valeur réelle. Nous associons donc non pas un marqueur binaire d'existence du calcul à sauvegarder, mais un coefficient d'approximation de celui-ci. Ainsi un calcul peu précis ne sera pas utilisé par un outil qui requiert un calcul plus précis. Le calcul effectué par cet outil remplacera alors l'ancien dans le gestionnaire de données.

Dans notre mise en œuvre, nous avons pris le parti de différencier les données utilisées fréquemment, en leur donnant un emplacement mémoire statique au sein du gestionnaire, et les données spécifiques allouée dynamiquement. Ceci permet un accès direct aux données les plus utilisées tout en ayant une structure ouverte, non contrainte dans le nombre de données stockées. Voici les données que nous avons considérées comme fréquentes :

- **Radiance** : estimation de la valeur de radiance émise par l'émetteur vers le récepteur. Les valeurs de minimum et maximum peuvent être aussi stockées pour les critères de raffinement utilisant des bornes d'erreur.
- **Transmittance** : estimation du facteur de visibilité entre les éléments. Comme pour la radiance les bornes minimales et maximales peuvent aussi être stockées.
- **Échantillons** : liste d'échantillons sur l'émetteur ou le récepteur.
- **Volume de vision** : une fois créé, le volume de vision est stocké, qu'il soit étendu ou non (cf. §1.2, page 99).
- **Axe** : l'axe centre-centre entre les éléments est utilisé par une grande partie des outils. Nous l'avons stocké (ainsi que sa longueur) malgré sa simplicité de calcul.
- **Aires projetées** : aires projetées de l'émetteur et le récepteur dans la direction de l'axe.
- **Obstacles potentiels** : les volumes de visions permettent de connaître une liste des obstacles potentiels. Cette liste permet d'optimisation les intersections avec la scène (calcul de facteur de forme et de facteur de visibilité par échantillonnage).

3.2 Paramétrisation.

Une grande majorité des outils dépendent de un ou plusieurs paramètres, tels que la précision requise par l'utilisateur, une plage de fonctionnement, etc. Ces paramètres permettent de moduler le comportement d'un oracle dont le schéma général reste celui de son graphe de décision. Ils peuvent être calculés de deux façons différentes :

- **Paramètres fixés** : les paramètres ne sont évalués qu'une seule fois au début du processus de raffinement. Il peuvent être directement manipulables grâce à une interface graphique ou fixés

par l'utilisateur. La paramétrisation de certains outils de l'oracle permet un contrôle manuel précis à différents niveaux.

- **Paramètres calculés** : les paramètres sont calculés à chaque exécution de l'outil par une méthode définie par le concepteur du raffineur. Ceci permet de faire varier la paramétrisation au cours du processus de raffinement, donc d'adapter localement et automatiquement le raffinement aux spécificités de la scène ou au raffinement déjà effectué.

3.3 Avantages de la structure utilisée.

La structure de graphe et la puissance de l'encapsulation de la programmation objet possèdent intrinsèquement de nombreux avantages, voici ceux que nous exploitons principalement :

- **Modularité** : tous les outils répondent à un même schéma de conception, ce qui guide la création des nouveaux outils. Cela permet de créer un oracle de raffinement comme un jeu de construction composé d'un ensemble de briques de base.
- **Évolutivité** : un des plus gros avantages de la modularité et de pouvoir ajouter à foison de nouvelles fonctionnalités, c'est par exemple le principe recherché pour pouvoir ajouter des outils fonctionnant sur l'aspect temporel de la radiosité. La seule caractéristique requise est de répondre à une question ou effectuer une action portant sur deux éléments dans une scène donnée.
- **Statistiques** : il est possible de comptabiliser le temps passé dans l'exécution de chaque noeud ou le nombre d'exécutions de ceux-ci. Ces statistiques permettent rapidement de déterminer les goulots d'étranglement d'un oracle de décision.
- **Réutilisation de branches** : une fois une branche de graphe construite et validée, elle peut-être réutilisée entière, telle un brique de base, dans un autre graphe de décision.
- **Vérification** : après chaque création de graphe de décision il est ainsi possible de vérifier que chaque noeud possède le bon nombre de paramètres et que chaque réponse est suivie par une question ou une décision. Les cycles peuvent être aisément repérés, ceci permet de garantir la validité du graphe.
- **Suivi d'exécution** : l'impression de chaque noeud et de sa réponse permet un suivi de chaque branche de raffinement au cours du processus de raffinement.
- **Recherche d'erreurs** : le suivi d'exécution associé au raffinement sélectif permet de trouver rapidement les causes d'un raffinement trop important ou trop faible.
- **Visualisation** : la visualisation d'une structure arborescente est très parlante pour l'humain, elle permet en un seul schéma intelligible de représenter un oracle complexe.
- **Génération d'interface** : chaque outil possédant sa propre interface liée à ses paramètres, il est aisé de générer automatiquement une interface graphique en fonction des outils utilisés dans le graphe.

Le processus d'exécution du graphe est un peu plus long que celui d'un raffineur classique (cf. §3.4, page 76) car c'est un processus récursif. Il demande, à chaque exécution d'un noeud, la recherche dans sa table de descendants du prochain noeud à exécuter. Il est heureusement possible d'établir une version itérative de l'algorithme. Ceci est réalisé par un générateur de code automatique qui crée un nouveau raffineur contenant la version "mise à plat" du processus de raffinement.

3.4 Validation et temps d'exécution.

La plate-forme utilisée pour le développement de cette structure est un logiciel de radiosité interne au laboratoire GRAVIR, dont le critère de raffinement habituel jusqu'alors utilisé était le critère BF. Le premier travail de validation de la structure de graphe de décision à été de recréer l'existant. Nous

présentons le graphe correspondant au critère BF (fig. 4.4), ainsi qu'une étude des temps d'exécution du graphe, de la version itérative générée et de l'ancien critère.

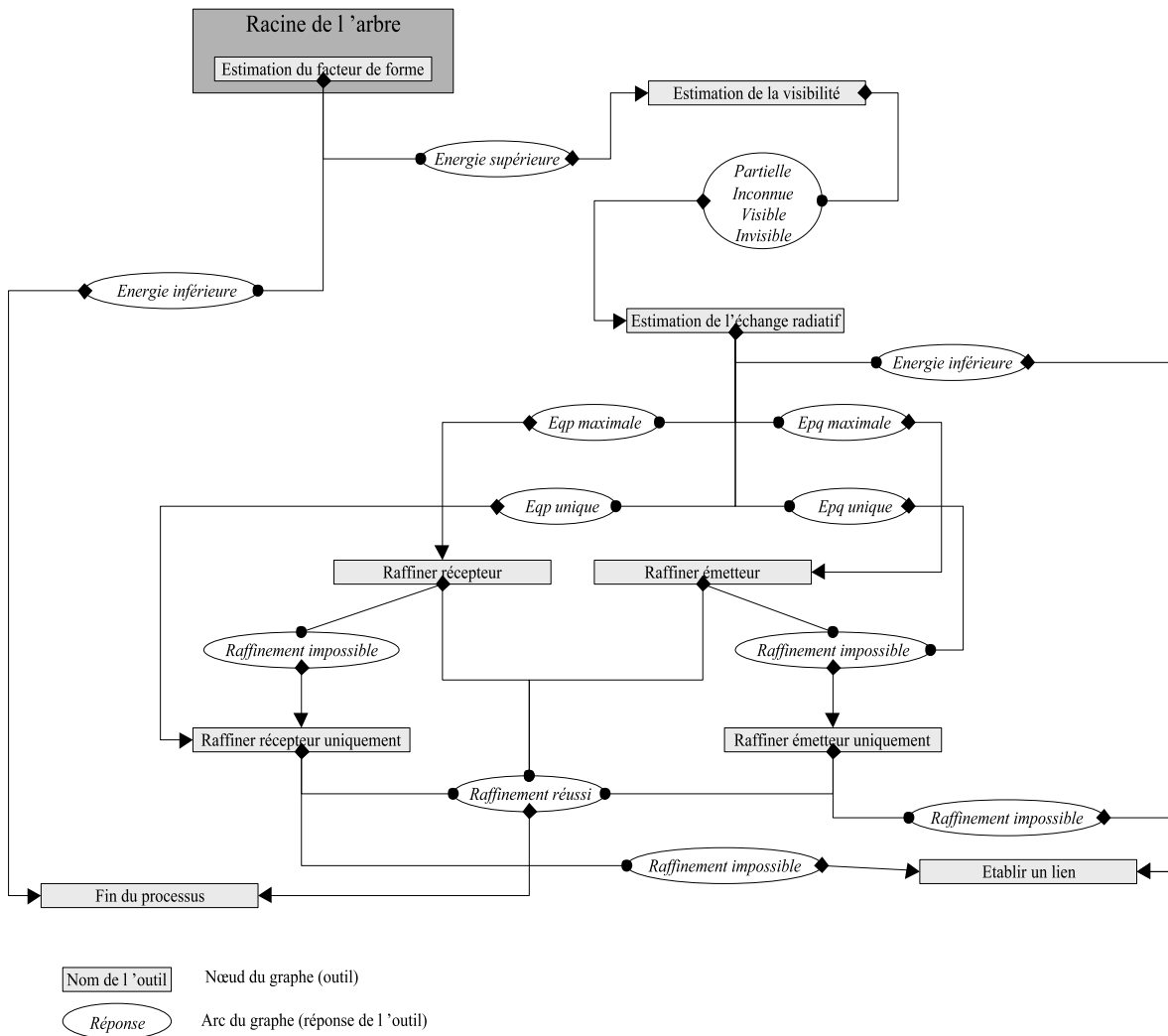


FIG. 4.4: Graphe de décision du critère BF.

Les scènes de test utilisées possèdent chacune leurs caractéristiques :

- **Room** : scène d'intérieur très simple ne possédant que très peu de problèmes de visibilité ;
- **TD** : scène d'intérieur constituée d'objets de tailles hétérogènes. Visibilité moyenne ;
- **Maze** : petite scène de labyrinthe, facteur de visibilité très élevé ;
- **Soda** : scène complexe d'intérieur, facteur de visibilité assez élevé ;
- **Séjour** : scène complexe d'intérieur avec des objets de tailles hétérogènes et un facteur de visibilité élevé.

Le nombre de polygones donné est le nombre de polygones avant raffinement.

On remarque que la perte de temps est relative à la complexité de la scène, l'écart se réduisant progressivement pour obtenir des augmentations de l'ordre de 10% dans la version récursive et quasiment nulles pour la version itérative. Par contre pour des scènes de faible complexité, le temps induit

	Room	TD	Maze	Soda	Séjour
Nombre de polygones	397	5260	313	77286	53875
BF	0,70 s.	2,30 s.	24,47 s.	83,70 s.	164,23 s.
version réursive	1,07 s. (+53%)	3,40 s. (+48%)	43,50 s. (+78%)	93,80 s. (+12%)	183,90 s. (+12%)
version itérative	0,80 s. (+14%)	2,47 s. (+7%)	26,47 s. (+8%)	84,27 s. (+1%)	167,490 s. (+2%)

TAB. 4.1: Comparaison des temps de calculs des versions itératives, récur-sives et normales d'un raffineur utilisant le critère BF.

par la structure n'est plus négligeable devant le temps propre du processus de raffinement. Notre but étant de travailler avec des scènes réalistes complexes, le très faible surcoût n'est pas pénalisant (tab. 4.1).

Les résultats visuels et quantitatifs obtenus par la structure du graphe de décision, comparés à ceux obtenus par le critère initial, sont strictement identiques et valident intégralement la fonctionnalité de celle-ci.

4 Outils.

La structure de graphe proposée découpe le raffinement en sous-tâches. Chacune de ces tâches élémentaires doit répondre à une unique question posée par l'oracle de raffinement.

Nous allons décrire les outils qui nous ont paru les plus importants tout au long de l'expérience acquise sur le processus de raffinement, ce qui représente une petite moitié de l'ensemble des outils développés. Nous les présenterons principalement par : la question à laquelle ils sont censés répondre ; les réponses possibles ; ainsi qu'une brève description de la technique utilisée.

4.1 Outils géométriques.

Portée.

L'outil de portée définit la position du récepteur par rapport à la géométrie de l'émetteur. Il est une généralisation de l'algorithme d'élimination de faces arrières (*backface culling*) à tous les éléments hiérarchiques (polygones, spots, groupes de surfaces, etc...).

Question :

- "Quelle est la position du récepteur par rapport à la géométrie du récepteur ?"

Réponses :

- Interne : l'ensemble du récepteur se trouve dans la portée de l'émetteur.
- Externe : l'ensemble du récepteur se trouve à l'extérieur de la portée de l'émetteur. L'échange radiatif est nul.
- À cheval : le récepteur intersecte la frontière de la portée de l'émetteur.

Paramètres :

- \emptyset
-

Cet outil se veut être un outil rapide qui trouvera sa place en haut du graphe de décision. Notre implémentation utilise la classification binaire de l'ensemble des points caractéristiques du récepteur par rapport à la portée de l'émetteur. Certaines erreurs peuvent intervenir dues à ce sous-échantillonnage mais restent très rares.

Angle solide.

L'outil d'angle solide estime l'angle solide entre l'émetteur et le récepteur et retourne la comparaison de cet angle avec une valeur seuil pour l'émetteur et le récepteur.

Question :

- "Pour quel élément l'angle solide est-il plus important que le seuil ?"

Réponses :

- Émetteur : l'émetteur est porteur du plus grand angle solide.
- Récepteur : le récepteur est porteur du plus grand angle solide.
- Aucun : le plus grand angle solide est plus petit que le seuil fixé.

Paramètres :

- Seuil : Angle solide minimal.
-

Nous utilisons un estimateur simple de l'angle solide considérant que les deux éléments se projettent en un disque dans toutes les directions (fig. 4.5).

on considère $A_1 < A_2$.

$$r_x = \sqrt{\frac{A_x}{\pi}}$$

$$l_1 = \frac{l \cdot r_1}{r_2 - r_1}$$

$$\alpha \simeq \frac{\pi r_1^2}{l_1} \simeq \frac{\pi r_1 (r_2 - r_1)}{l}$$

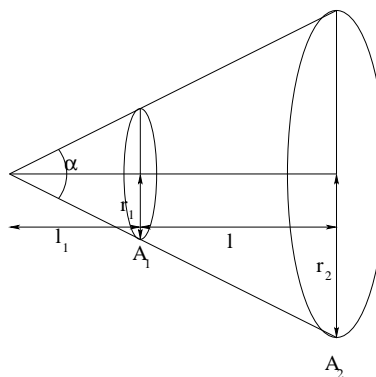


FIG. 4.5: Estimation de l'angle solide sous-tendu par deux éléments dont leurs projections sont considérées circulaires.

Estimation de volume.

Cet outil calcule une estimation du volume de l'espace entre l'émetteur et le récepteur (volume de vision (cf. §1.2, page 99)).

Question :

- "Le volume entre les deux éléments est-il plus ou moins grand que le seuil ?"

Réponses :

- Supérieur
- Inférieur

Paramètres :

- Volume minimal : valeur seuil du volume comparée avec le volume estimé par l'outil.

Superposition.

Cet outil détermine la position relative des deux boîtes englobantes des éléments hiérarchiques.

Question :

- "Quelle est la position relative des boîtes englobantes ?"

Réponses :

- Disjointes : les deux boîtes englobantes sont entièrement disjointes.
- Égales : les deux boîtes englobantes sont identiques.
- Intérieure : la boîte englobante du récepteur est à l'intérieur de celle de l'émetteur.
- Englobante : la boîte englobante du récepteur englobe celle de l'émetteur.
- superposées : les boîtes englobantes s'interpénètrent.

Paramètres :

- \emptyset

Classification par distance.

Le traitement d'éléments proches ou très éloignés peut-être très différent. Cet outil peut-être utilisé pour appliquer une des règles utilisée depuis presque un siècle en ingénierie de l'éclairage : *the five time rule* [Ash94]. Nous avons utilisé un outil qui répartit les distances minimales en trois classes : proches, loin et entre les deux ; en fonction de deux seuils.

Question :

- "Quelle est la classification de distance entre les deux éléments ?"

Réponses :

- Proche : la distance est inférieure au seuil minimal.
- Entre-Deux : la distance est comprise entre les deux seuils.
- Éloigné : la distance inter-éléments est plus grande que le seuil maximal.

Paramètres :

- Deux seuils : valeurs minimale et maximale de distance.

Cet outil est archaïque dans ses réponses. Dans certains cas, il pourrait être plus intéressant d'obtenir une classification dépendante de la taille relative des éléments et non pas une qui soit uniquement basée sur la distance entre les centres.

Distribution de distances.

La valeur du facteur de forme peut-être estimée plus ou moins bien (ou rapidement) en fonction de la disposition géométrique des éléments. Lorsque la fonction de distance est très hétérogène une estimation simple du facteur de forme risque d'être erronée. Il est intéressant de connaître la distribution de distances entre les éléments afin de savoir quel algorithme de calcul de facteur de forme utiliser ou si il n'est pas utile d'essayer de l'estimer.

Question :

- *"La distribution de distance est-elle homogène entre les deux éléments ?"*

Réponses :

- Homogène : la distribution de distance est inférieure au seuil défini.
- Hétérogène : la distribution est supérieure.

Paramètres :

- Seuil de distribution : distribution minimale de distance.
-

La distribution de distances peut se calculer plus ou moins précisément. En utilisant un échantillonnage des deux éléments on peut calculer la distribution comme la variance ou l'étendue des distances.

Il est aussi possible d'utiliser des bornes moins précises telles qu'une comparaison de la distance aux centres et des tailles estimées des éléments.

4.2 Outils énergétiques.

Estimation de la variance.

La fonction d'échange radiatif point à point n'est pas constante entre les éléments. Cet outil estime sa variance et détermine quel est l'élément qui contribue le plus à cette variation.

Question :

- *"La variance de la radiance est-elle suffisamment uniforme entre les éléments ?"*

Réponses :

- Uniforme : aucune des variances calculées n'est supérieure au seuil fixé.
- Récepteur non uniforme, émetteur non uniforme : seule la variance sur l'élément (récepteur ou émetteur) est supérieure au seuil.
- Récepteur et émetteur non uniforme, émetteur et récepteur non uniforme : les deux variances des éléments sont supérieures au seuil, dans l'ordre indiqué.

Paramètres :

- Seuil : seuil de variance minimal.
-

Gain.

Le gain est l'estimation de l'apport de l'échange énergétique en fonction de la radiosité déjà présente sur le récepteur. La radiosité déjà présente peut être directement la radiosité locale mais peut aussi être celle de niveaux hiérarchiques supérieurs plus globaux.

Il existe une multitude de façon d'estimer le gain énergétique apporté suivant que l'on utilise des fonctions linéaires, logarithmiques, perceptives, etc. La question et les réponses sont quasiment identiques, seule la méthode utilisée varie.

Question :

- "L'apport énergétique est-il suffisant ?"

Réponses :

- Indéfini : la radiosité locale est nulle.
- Supérieure : la valeur de gain est supérieure au seuil.
- Inférieure : la valeur de gain est inférieure au seuil.

Paramètres :

- Seuil : valeur de gain minimale.

4.3 Outils d'action.

Les outils d'actions sont un peu particuliers car ils ne répondent pas à une question mais exécutent une action sans condition. Les décisions du graphe sont les *outils de décision*, ils se trouvent obligatoirement aux extrémités (feuilles) du graphes. Les autres types d'actions sont soit des sélecteurs, qui déterminent un type de donnée ou d'objet, soit des outils de débogage, de visualisation ou d'aide à la compréhension du processus.

Outils de subdivision.

Le raffinement d'un lien est une des décisions possibles de l'oracle. Pour ce faire on subdivise un ou les deux éléments hiérarchiques du lien. Nous n'utilisons qu'un seul outil prenant en paramètre les éléments à subdiviser.

Si l'on utilise un critère de subdivision maximale afin de limiter la précision de la solution, il est possible que l'action ne puisse pas aboutir entièrement. L'outil retourne l'état de son action.

Les deux seuls véritables outils de décision – au sens premier du terme – sont les outils d'établissement de lien et de suppression de lien (ou de non-établissement s'il n'existe pas encore). L'outil de raffinement n'étant qu'une étape directement préliminaire de la décision. Toutefois, nous l'avons classifié comme tel car s'il réussit, il est obligatoirement un élément terminal.

Action :

- "raffinement du lien par la subdivision d'un ou des deux éléments hiérarchiques le composant"

Réponses :

- Éléments subdivisés : les deux éléments ont été subdivisés.
- Émetteur subdivisé : l'émetteur seul a été subdivisé.
- Récepteur subdivisé : le récepteur seul a été subdivisé.
- Échec : aucun élément n'a pu être subdivisé.

Paramètres :

- Deux indicateurs booléens correspondants à l'émetteur et au récepteur.

Suppression d'un lien.

Lorsque l'oracle de raffinement se rend compte définitivement qu'un lien n'a pas lieu d'être (c'est-à-dire lorsqu'il est persuadé que l'influence de ce lien ne pourra jamais être autre chose que nulle sur la solution de radiosité), il décide de supprimer ce lien des échanges radiatifs possibles. On parle souvent de "suppression" mais ce n'est, à proprement parler, que le fait de ne pas créer de lien entre les éléments traités, car les liens sont créés a posteriori.

Décision : *"Supprime le lien de la liste des liens ayant une influence sur la solution de radiosité."*

Pas de réponse : Élément terminal

Paramètres :

- \emptyset

Établissement d'un lien.

Si l'approximation de l'échange radiatif semble suffisant le lien est établi au niveau hiérarchique courant. Les données nécessaires sont le facteur de forme et une estimation du facteur de visibilité entre les deux éléments. Si ces valeurs n'existent pas ou ne sont pas suffisamment précises, l'outil peut les recalculer.

Décision : *"Établit le lien entre les deux éléments hiérarchiques courants."*

Pas de réponse : Élément terminal

Paramètres :

- \emptyset

Calcul du facteur de forme.

Dans la théorie, le facteur de forme intègre la visibilité. Dans l'algorithme de radiosité il est fréquent de découpler les deux. L'outil calcule par échantillonnage le facteur de forme en intégrant ou non la visibilité en fonction d'un paramètre utilisateur.

Action :

- *"Calcule le facteur de forme par échantillonnage."*

Réponses :

- calcul de facteur de forme exécuté

Paramètres :

- Nombre d'échantillons sur l'émetteur.
 - Nombre d'échantillons sur le récepteur.
 - Calcul intégrant la visibilité ou non.
-

Calcul du facteur de visibilité par échantillonnage.

L'évaluation du facteur de visibilité est relativement complexe, il est une partie importante du processus de raffinement. Actuellement, le seul outil d'évaluation que nous utilisons est une estimation par échantillonnage.

Lorsque l'on utilise un algorithme de radiosité par propagation, il est possible de recourir à des algorithmes plus coûteux en temps mais qui factorisent un nombre d'informations réutilisables pour l'émetteur important, tel que le tampon de profondeur ou son évolution, méthode hybride décrite par l'équipe de l'INRIA de Nancy [HA00]. Nous resterons dans le cadre d'une évaluation de visibilité ponctuelle avec une connaissance très faible de la scène environnante.

Action :

- *"Calcule le facteur de visibilité par échantillonnage."*

Réponses :

- calcul de visibilité exécuté

Paramètres :

- Nombre d'échantillons sur l'émetteur.
 - Nombre d'échantillons sur le récepteur.
-

Sélecteurs.

Un sélecteur est un outil qui permet de caractériser une donnée qui ne relève ni de la géométrie, ni de l'énergie, ni de la visibilité. Ce type d'outil se situe fréquemment en début de graphe afin d'appliquer un raffinement spécifique – une branche de graphe – à une situation spécifique. Nous allons décrire ici le sélecteur de type de source lumineuses.

Le raffinement est très différent en fonction du type de source lumineuse qui est traité. L'éclairage dû au ciel, au soleil, à une source ponctuelle ou à une source étendue est très différent car il peut produire des ombres nettes ou très douces. Le sélecteur de source lumineuses est un outil très utile.

Action :

- "Quel est le type de source lumineuse de l'émetteur ."

Réponses :

- Source ponctuelle.
- Source étendue.
- Source volumique.
- Source directionnelle.
- Soleil.
- ciel.

Paramètres :

- \emptyset

Outils de débogage.

Les outils de débogage sont les outils qui permettent de connaître localement le comportement de l'oracle. Ils sont complémentaires des statistiques générées par le graphe afin de comprendre à la fois globalement et précisément la trame du raffinement.

Nous n'allons pas détailler les nombreux outils que nous avons utilisés car la recherche d'erreur est très dépendante de ce qui est cherché. Nous nous bornerons à proposer deux outils utiles :

- Erreur : lance une erreur définie et stoppe le processus de raffinement. Permet de vérifier qu'une branche du graphe n'est effectivement pas utilisée si elle n'est pas prévue pour. L'outil de lancement d'erreur est une feuille particulière du graphe.
- Sauvegarde : sauvegarde l'ensemble des éléments liens passant par la branche de graphe. Ceci permet de réitérer un même raffinement (ou une partie du raffinement) plusieurs fois grâce à un outil de lecture de liens sauvegardés. Grâce à cet outil, il est aussi possible de se rendre compte visuellement de l'ensemble des éléments impliqués dans une branche de raffinement.

4.4 Outils de visibilité.

Dans notre optique, les outils de visibilité se doivent de déterminer la classification de visibilité entre les deux éléments extrémités d'un lien. Ils ne calculent pas forcément une estimation du facteur de visibilité mais peuvent le faire, cette valeur sera alors stockée dans le gestionnaire de données.

Les outils ci-dessous décrits utilisent les notions définies dans la section visibilité (cf. chapitre 5, page 97).

Classification par échantillonnage.

La classification de la visibilité par échantillonnage est une évaluation à partir de la classification d'un certain nombre de rayons. Le nombre de rayons est défini par un paramètre.

La valeur du facteur de visibilité se calcule facilement comme le nombre de rayons visibles sur le nombre de rayon total lancé. Par contre, la classification stoppant ses investigations dès que deux rayons sont classifiés différemment – la classification PARTIEL est certifiée – la valeur du facteur de visibilité est alors incomplètement déterminée. Ce qui engendre une valeur erronée si on la considère comme une bonne approximation. Au cours de notre étude, nous avons décidé de ne plus nous fier à cette valeur qui génère un bruit trop important.

Question :

- *"Classification de la visibilité par échantillonnage de l'émetteur et du récepteur."*

Réponses :

- **PROBABLEMENT VISIBLE** : aucun rayon n'a été occulté.
- **PARTIEL** : deux rayons sont classifiés différemment.
- **PROBABLEMENT INVISIBLE** : tous les rayons ont été occultés.

Paramètres :

- Nombre d'échantillons de l'émetteur.
- Nombre d'échantillons du récepteur.
- Nombre maximal de rayons : une alternative serait de fixer un temps maximal imparti.

Classification par volume de vision.

Les volumes de vision permettent de classier exactement la visibilité totale contrairement à l'échantillonnage par rayons. Il est aussi possible de déterminer la visibilité nulle créée par un obstacle unique (cf. §1.3, page 103). Ceci leur confère un rôle très important malgré leur coût en temps.

Nous étudions dans le chapitre de la visibilité la méthode d'accélération de la classification de la visibilité par volumes de vision ainsi qu'une optimisation de leur génération (cf. §1.3, page 103).

L'outil de volume de vision peut répondre à une précision déterminée, en prenant uniquement en compte des obstacles d'une certaine taille. Notre expérience nous dirige plutôt vers une classification exacte, les approximations de la visibilité étant déjà élevées et très influentes sur le raffinement.

Question :

- *"Classifie la visibilité entre l'émetteur et le récepteur"*

Réponses :

- **Visibilité TOTALE** : aucune boîte englobante des plus petits obstacles potentiels n'empiète sur le volume de vision.
- **Visibilité PROBABLEMENT PARTIEL** : une boîte englobante intersecte le volume de vision.
- **Visibilité INVISIBLE** : un obstacle convexe intersecte toutes les arêtes longitudinales du volume de vision.

Paramètres :

- Taille d'obstacle minimale.

5 Usage de la structure.

Dans cette section, nous allons donner un exemple d'utilisation de la structure de graphe définie précédemment. Nous, allons, grâce à l'utilisation des différents outils de visibilité et à l'étude des statistiques générées par la structure de graphe, déterminer différents problèmes rencontrés et leur apporter une solution.

Dans l'ensemble des paragraphes suivants, nous schématiserons les oracles uniquement par leurs outils principaux.

5.1 Oracle initial.

Nous allons partir du critère de raffinement BF déjà décrit précédemment pour la validation de la structure (cf. §3.4, page 76). Il est composé de trois noeuds principaux : une estimation de facteur de forme, une évaluation de la classification et du facteur de visibilité, puis un critère décisionnel simple : un seuil sur l'irradiance.

L'avantage du critère BF est la simplicité des calculs effectués, cela lui permet d'être très rapide et stable dans une majorité des configurations. La visibilité est prise en compte en modifiant le seuil sur l'irradiance par un facteur défini par l'utilisateur. La visibilité nulle ne peut pas être garantie, ce qui l'empêche de stopper le raffinement dans des zones où ce n'est pas nécessaire.

Les désavantages de ce critère simpliste sont nombreux dès que l'on rencontre des configurations particulières. Le plus gros défaut est la difficulté de contrôle de l'utilisateur sur la solution. Le seuil, fixé manuellement est absolu dans la gamme d'intensité lumineuse. Ainsi, il faut connaître la luminosité d'une scène pour pouvoir fixer ce paramètre.

Le seuil est constant sur toute la gamme d'intensité quelle que soit la luminosité ambiante. Donc l'écart perceptible par l'oeil est considéré identique pour toutes les intensités alors que la courbe d'adaptation de l'oeil à son environnement est principalement logarithmique.

La même estimation du facteur de forme est effectuée quelle que soit la configuration géométrique. En particulier lorsque l'émetteur est très proche d'un grand récepteur l'approximation faite peut engendrer la suppression de liens importants.

Même pour des obstacles de taille importante, la visibilité est effectuée par échantillonnage. Si tous les rayons sont occultés, par malchance ou parce qu'un obstacle est très proche de l'émetteur, le lien risque d'être supprimé alors que l'énergie restante peut être importante. Le risque majeur est, du fait d'un sous échantillonnage, de ne pas être cohérent dans les décisions prises d'un élément hiérarchique à ses voisins et ainsi créer des discontinuités.

Dans l'exemple que nous utilisons, le critère BF n'a pas réussi à déterminer le passage lumineux dû à la porte entre les deux pièces avec un échantillonnage de 4 rayons (fig. 4.6). L'étude du lien entre les deux pièces révèle deux anomalies : la visibilité est classifiée *PROBABLEMENT INVISIBLE*, et l'échange énergétique est considéré trop faible d'une cellule à une autre. Ceci est dû aux approximations faites sur l'émission des cellules, les sources sont perpendiculaires à la direction entre les deux cellules.

Dans cet exemple, il faut monter à une valeur d'échantillonnage entre 32 et 64 rayons afin que l'oracle prenne en compte la porte. Nous avons ajouté un noeud qui caractérise la variation d'intensité de l'émetteur afin de subdiviser les liens dont l'émetteur n'est pas suffisamment uniforme. Ainsi fait, le lien reliant les deux pièces (fig. 4.7) est subdivisé.

Pour des pinces très fins, un échantillonnage de 64 rayons est trop important en considérant le rapport temps/qualité. Dans la prochaine section nous allons utiliser un échantillonnage adaptatif en fonction de la taille du pinceau.

5.2 Échantillonnage adaptatif.

Nous considérons qu'un échantillon est nécessaire par élément de surface minimal fixé par l'utilisateur. Ainsi, le nombre d'échantillons est proportionnel à la taille de chaque élément, limité à la valeur de 64, valeur fixée expérimentalement pour cette scène. Le nombre de rayons est égal au maximum des deux échantillonnages.

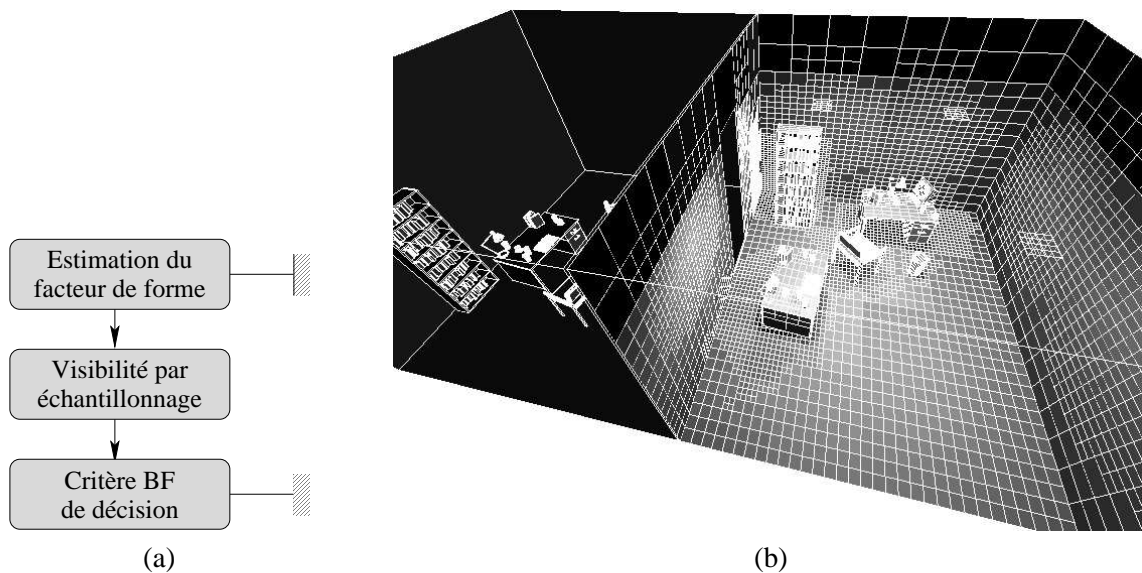


FIG. 4.6: Solution de radiosité utilisant le critère BF (a) sur la scène des deux bureaux (b). Le lien transitant d'une pièce vers la seconde n'est pas subdivisé.

La subdivision est beaucoup trop importante dans les zones d'ombre totale car la visibilité avec la pièce non éclairée est toujours soit **PROBABLEMENT INVISIBLE** soit **PARTIEL**. Dans le premier cas, nous ne pouvons pas croire la réponse et dans le second, la subdivision est incitée. Dans les deux cas la subdivision est probable (fig. 4.7).

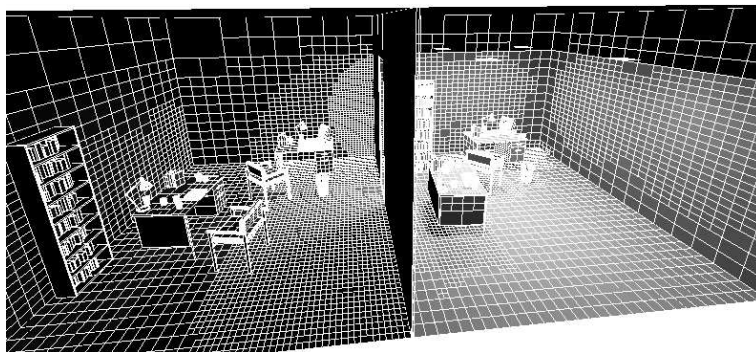


FIG. 4.7: Sur-raffinement de la solution dans les zones de visibilité nulle ou partielle avec l'utilisation d'un outil de classification par échantillonnage.

On remarque, dans l'analyse des réponses de l'outil de visibilité, qu'une proportion non négligeable de classification **INVISIBLE** est déterminée lors de la première itération (16%) (tab. 4.2). Il serait intéressant d'utiliser un outil permettant de caractériser exactement cette classification. Dans la prochaine section, nous allons ajouter l'utilisation de volumes de vision étendus.

	1 ^{ère} itération	2 ^{ème} itération
Temps de raffinement total	: 72,8 s.	991,9 s.
Nombre de liens	: 247.987	1.387.100
Outil d'échantillonnage :		
Temps relatif	: 70,0 %	60,7 %
Temps par processus (μ s)	: 164	672
PROBABLEMENT VISIBLE	: 34%	21%
PROBABLEMENT INVISIBLE	: 16%	22%
PARTIEL	: 50%	57%

TAB. 4.2: Statistiques du raffinement pour le raffineur utilisant une le critère BF et une classification de visibilité par échantillonnage adaptatif.

5.3 Visibilité gérée par volumes de vision étendus.

En remplaçant le noeud de classification de la visibilité du graphe initial par un outil utilisant les volumes de vision étendus, le raffinement repère la porte entre les deux pièces, et la subdivision de fait au plus juste. Cela permet une simulation beaucoup plus proche de la réalité mais le coût est important. Les temps de calcul sont plus que doublés passant de 73 s. à 181 s. (tab. 4.3).

On peut remarquer que, à la première itération, la pièce non éclairée n'est pas subdivisée là où le mur cache complètement les sources lumineuses (fig. 4.8). Parce que l'utilisation de volumes de vision est coûteuse, nous allons voir dans la section suivante, un graphe permettant de réduire le nombre d'appels de l'outil utilisant les volumes de vision.

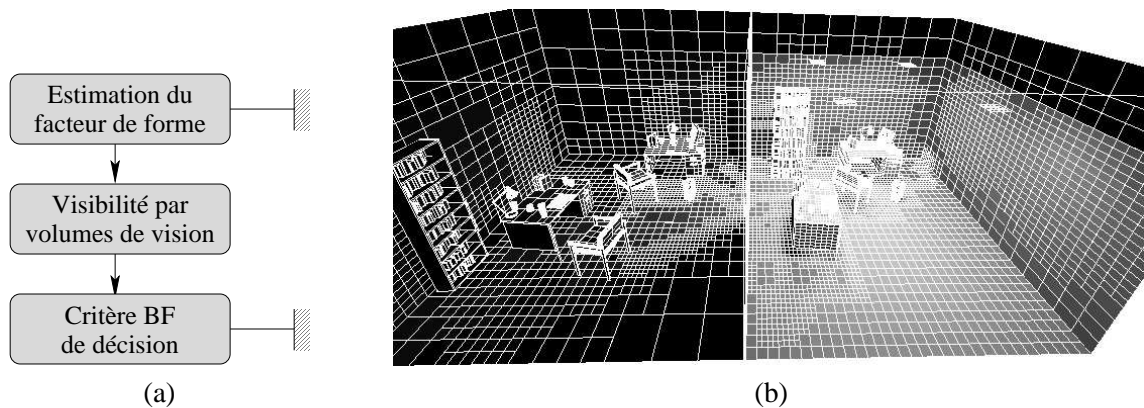


FIG. 4.8: Solution de radiosité utilisant une classification de visibilité par volumes de vision (a). Les zones complètement occultées pour l'éclairage direct n'ont pas été subdivisées (b).

5.4 Visibilité combinée.

Nous allons utiliser une évaluation de la visibilité mélangeant les outils d'échantillonnage et de volumes de vision étendus (cf. §1.3, page 103). Le nombre de rayons est toujours limité à 64, cette valeur est atteinte pour des éléments de surface ayant une aire 64 fois plus importante que la taille minimale d'aire fixée.

	1 ^{ère} itération	2 ^{ème} itération
Temps de raffinement total	: 181,1 s.	1125 s.
Nombre de liens	: 195.146	1.576.429
Outil de volumes de vision :		
Temps total	: 77,3%	72,7%
Temps par processus (μ s)	: 282	706
PROBABLEMENT VISIBLE	: 12%	7%
PROBABLEMENT INVISIBLE	: 5%	10%
PARTIEL	: 83%	83%

TAB. 4.3: Statistiques du raffinement pour le raffineur utilisant une classification de la visibilité uniquement par volumes de vision.

Nous allons utiliser cette limite pour choisir entre les deux outils de visibilité. Les liens dont la section maximale est inférieure utiliseront une classification par échantillonnage, les autres une classification par volumes de vision étendus. Le graphe de raffineur correspondant peut être de deux formes : la première en plaçant un noeud de sélection qui utilise un critère sur l'aire des éléments ; la seconde en utilisant la réponse de l'outil de classification par échantillonnage de dépassement du nombre de rayons autorisé. Nous utilisons cette deuxième méthode parce que l'information de taille maximale des aires projetées est déjà calculée dans l'outil d'échantillonnage, ce qui permet de ne pas effectuer de calculs redondants (fig. 4.9a).

En utilisant cette technique d'échantillonnage des éléments, la taille du plus petit obstacle "raté" est de l'ordre de la taille limite d'aire. Nous considérons que ces obstacles sont suffisamment petits pour être visuellement négligeables et ainsi nous permet de considérer la classification comme certaine.

Cette approximation n'est pas valable dans le cas de petits objets très proches du récepteur, où l'ombre produite sera perceptible. Hélas, nous n'avons pas d'estimateurs de la distance des obstacles par rapport aux émetteurs et aux récepteurs. Un tel estimateur nécessiterait de connaître l'ensemble des obstacles qui ne sont pas entièrement en dehors du pinceau reliant l'émetteur et de récepteur.

On peut remarquer que le nombre de liens établi est le plus faible de toutes les méthodes, car la subdivision est réalisée à partir de classifications certaines ou quasi-certaines. Le maillage résultant correspond effectivement relativement bien aux variations de la fonction de radiosité.

On peut voir que l'outil de classification par volume de vision est très coûteux, jusqu'à 8 fois plus que l'outil par échantillonnage dans la deuxième itération, mais les temps de calculs de la solution sont les plus faibles. Effectivement, le temps utilisé à la détermination de la visibilité nulle est compensé par la simplification du maillage résultant (tab. 4.4).

5.5 Conclusion.

Grâce à la souplesse de la structure de raffinement et à la génération automatique de statistiques, l'étude de critères de raffinement est grandement aidée. Nous avons présenté l'intégration de la visibilité au sein d'un critère simple, en nous faisant guider par l'étude des temps et de la répartition des réponses de chaque outil.

La structure de raffinement est aussi très utile pour comprendre une "erreur" de raffinement. Dans le cas du critère BF initial (cf. §5.1, page 87), un lien n'a pas été subdivisé, provoquant une erreur très importante sur la solution finale. La traçabilité du processus de raffinement sur ce lien particulier permet de déterminer efficacement la raison de cette erreur.

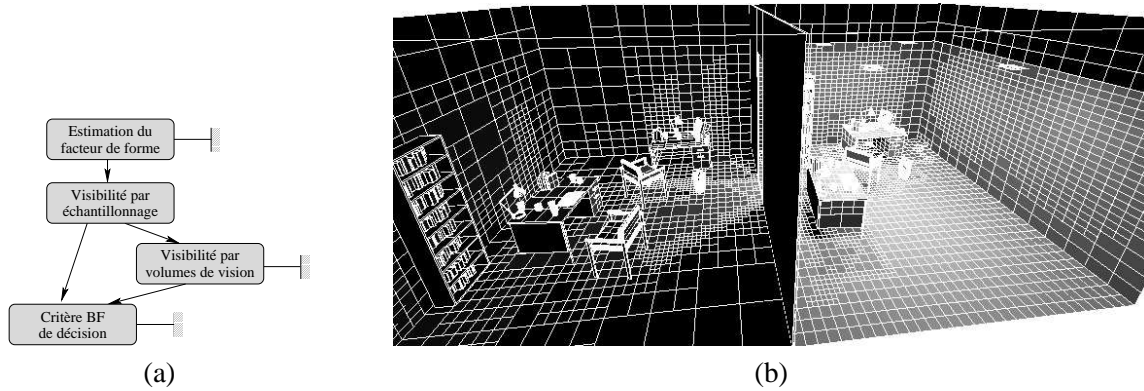


FIG. 4.9: Solution de radiosité utilisant une classification de visibilité par un mélange de volumes de vision et d'échantillonnage (a). Le maillage est bien adaptée aux variations de la fonction de radiosité (b).

	1 ^{ère} itération	2 ^{ème} itération
Temps de raffinement total	: 46,6 s.	972,1 s.
Nombre de liens	: 70.883	1.262.559
Outil d'échantillonnage :		
Proportion d'appel	: 94%	57%
Temps total	: 66%	13,4%
Temps par processus (μs)	: 154	137
PROBABLEMENT VISIBLE	: 33%	13%
PROBABLEMENT INVISIBLE	: 15%	15%
PARTIEL	: 49%	18%
Outil de volumes de vision :		
Proportion d'appel	: 2%	30%
Temps total	: 4%	55%
Temps par processus (μs)	: 611	1064
PROBABLEMENT VISIBLE	: 35%	8%
PROBABLEMENT INVISIBLE	: 3%	7%
PARTIEL	: 62%	85%

TAB. 4.4: Statistiques du raffinement pour le raffineur utilisant une classification de la visibilité uniquement par volumes de vision.

6 Raffinement collaboratif.

Durant ces trois années de recherches sur le raffinement, nous nous sommes confronté à une multitude de "petits" problèmes. Chaque configuration géométrique est différente, elles requièrent chacune une solution qui parfois provoque de nouvelles erreurs. Pourtant fréquemment, en regardant la solution créée par l'oracle, la solution est évidente pour l'oeil humain. On arrive pertinemment à savoir quel ombre est importante, quel détail doit être mis en valeur, quelles discontinuités sont désagréables à regarder, etc.

L'oracle de raffinement, malgré l'introduction de critères perceptifs simples, n'arrive pas à déterminer les facteurs importants dans une image. De plus les approximations faites lors des différents calculs ne sont pas bornées, ce qui empêche de différencier les calculs erronés des calculs justes.

Actuellement, dans les applications artistiques, les graphistes réalisent quasiment entièrement le travail de simulation d'éclairage en recréant artificiellement la propagation de la lumière. Leur compétence et leur œil exercé leur permet de donner l'illusion, mais c'est un travail long et approximatif.

L'ordinateur ne pouvant calculer une solution visuellement agréable et l'humain ne pouvant garantir de créer une simulation juste, nous proposons une collaboration entre les deux entités afin d'obtenir un résultat juste et agréable.

Nous ne présenterons dans cette partie qu'une ébauche d'une telle collaboration par un exemple d'amélioration d'ombres portées. Mais beaucoup de problèmes peuvent être facilement résolus par l'utilisateur, dont les suivants :

- Amélioration d'ombres : certaines ombres sont importantes dans une scène et l'algorithme de radiosité classique manque fréquemment de petits objets. Typiquement les pieds de chaise ont rarement une ombre précise.
- Repérage de discontinuités : les nombreuses approximations des échanges lumineux provoquent parfois des sauts de seuils pour des éléments contigus. Cela provoque des discontinuités désagréables à l'œil dans la solution de radiosité.
- Déraffinement : pour sauvegarder de la mémoire et améliorer la rapidité du rendu, un maillage minimal s'impose. L'utilisateur peut définir les zones de faible intérêt.

Dans notre première approche d'un raffineur collaboratif, nous avons opté pour une sélection des liens potentiellement modifiables par le biais de la géométrie de la scène. Le but étant de créer une solution correcte, le travail dans l'espace image ne nous a pas semblé justifié, la collecte terminale travaille déjà dans l'espace image de manière suffisante.

Il faut donc trouver une interface entre l'utilisateur et l'oracle de raffinement qui permette de déterminer précisément quels sont les liens impliqués dans la mise à jour désirée par l'utilisateur.

6.1 Détermination des liens.

Nous avons choisi d'utiliser trois listes de sélecteurs d'éléments correspondants aux émetteurs, récepteurs et obstacles pouvant intervenir dans les échanges énergétiques à modifier. Les liens que nous traiterons sont ceux qui interagissent avec les trois groupes d'éléments simultanément. Toutes les listes n'ont pas besoin d'être utilisées en même temps, on peut, par exemple, ne spécifier qu'un émetteur afin de reprendre l'ensemble des échanges radiatifs qu'il émet.

La détermination des liens à réévaluer se fait par un parcours de la hiérarchie de la scène. Chaque lien est comparé avec les trois groupes de sélection, à chaque groupe correspond un test particulier.

- Récepteur : en radiosité hiérarchique par collecte, les liens sont stockés sur le récepteur. Ainsi, pendant le parcours de la hiérarchie de la géométrie, on peut éliminer directement les branches qui ne contiennent pas un seul élément de la liste de sélection de récepteurs. Ceci permet d'éliminer très rapidement les branches entières de la hiérarchie qui ne pourront pas être influencées par les modifications.
- Émetteur : les tests avec le groupe de sélection d'émetteurs se fait pour chaque lien durant le parcours de la hiérarchie de la géométrie. Les liens dont l'émetteur n'appartient pas à la descendance du groupe de sélection d'émetteur ou dont aucun émetteur de sélection n'appartienne à la descendance de l'émetteur courant peuvent être éliminés de la liste des liens à mettre à jour.

- Obstacle : un volume de vision est lancé entre l'émetteur et le récepteur du lien courant. Sont éliminés de la liste de liens à mettre à jour, les liens dont aucun obstacle n'occulte le volume de vision.

L'information d'un récepteur, ou d'un ensemble de récepteurs, est la plus importante des trois. En effet, elle permet de réduire fortement le nombre de tests effectués en éliminant des branches entières de la hiérarchie.

Les liens restant dans la liste de mise à jour sont traités en fonctions du problème spécifié par l'utilisateur. Dans notre prémisses de mise en œuvre nous avons opté pour laisser le choix à l'utilisateur du raffineur à appliquer sur ces liens. S'il désire améliorer les ombres, il va utiliser un raffineur spécialisé dans la caractérisation des ombres et une subdivision minimale du maillage plus fine. S'il désire éliminer une discontinuité, il va avoir recours à un raffineur qui évalue le facteur de forme très précisément et tolèrera une erreur relativement faible.

Cette interface rudimentaire permet d'obtenir des résultats encourageants.

6.2 Exemple d'amélioration d'une ombre.

Nous allons utiliser un raffinement collaboratif sur une ombre importante d'une scène qui n'a pas été correctement traitée dans le raffinement initial. La corbeille à papier au milieu du chemin doit avoir une ombre précise car c'est l'objet central et il est le plus éclairé de la pièce (fig. 4.10a).

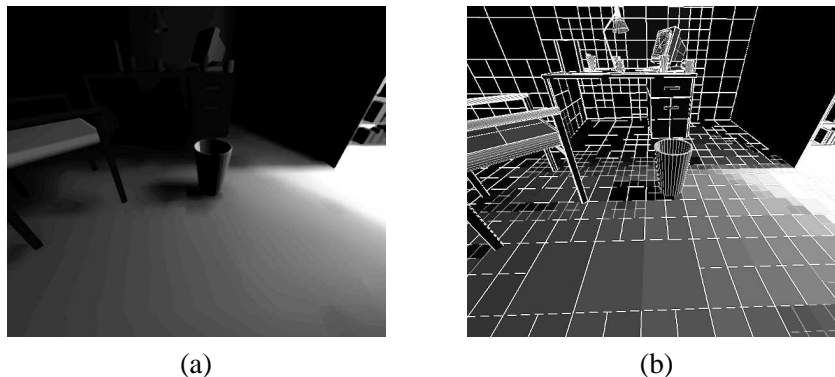


FIG. 4.10: Raffinement initial où les ombres des objets principaux ne sont pas bien définies.

Nous allons ensuite définir la zone où l'ombre doit être améliorée en définissant la zone où l'ombre doit se projeter comme un récepteur à mettre à jour. Nous pouvons aussi définir la corbeille comme le seul obstacle à traiter afin de limiter le nombre de calculs (fig. 4.11).

Nous appliquons un raffineur spécialisé dans la caractérisation des contours d'ombres pour effacer les discontinuités d'ombrage formées et obtenir une ombre précise (fig. 4.12).

Nous avons utilisé deux solutions initiales : la première est faiblement définie au niveau des échanges énergétiques et n'a été calculée qu'avec une itération. Peu de liens ont été créés (87.000) ; la seconde est plus détaillée au niveau énergétique, mais suit les mêmes paramètres au niveau de la visibilité. De plus le nombre d'itérations est de 4 ce qui permet d'obtenir beaucoup plus de liens (1.5million) pour une qualité équivalente des ombres au niveau de la corbeille qui nous intéresse.

Les temps de calculs dépendent de l'ampleur des mises à jour à produire. Lorsque le nombre de liens est important, le temps de conservation de la cohérence de la solution n'est plus négligeable et peut prendre plus de la moitié du temps de mise à jour. Cependant, en comparaison du temps de calcul de la solution initiale, cela reste un temps assez court (tab. 4.5).

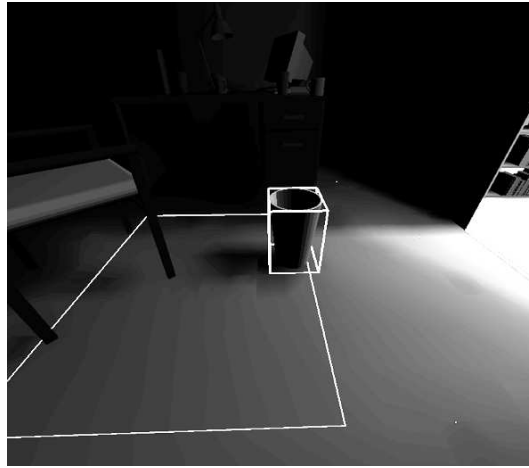
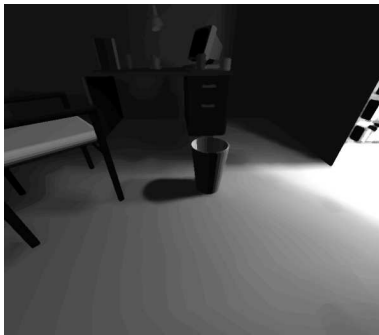
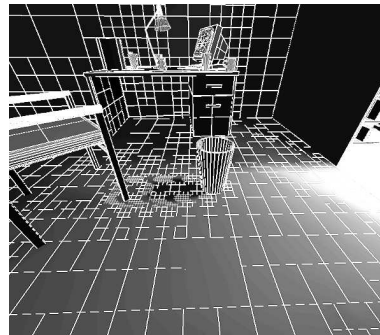


FIG. 4.11: Sélection d'un ensemble de récepteurs et d'un obstacles afin d'améliorer la solution de radiosité des liens étant impliqués avec eux.



(a)



(b)

FIG. 4.12: Raffinement précis des ombres dans les zones déterminées par l'utilisateur.

Le raffinement collaboratif est un palliatif à une heuristique trop complexe à identifier où à une puissance de calcul trop faible. L'intervention humaine permet de porter rapidement un jugement avec nos critères de perception sur la qualité de la solution, ce qu'une heuristique ne peut pas faire à peu de frais.

La collaboration est active dans le sens où l'utilisateur propose des améliorations et l'oracle de raffinement compose la solution afin de rester cohérent au niveau des échanges énergétiques.

7 Bilan.

L'étude des critères de raffinement n'est pas chose aisée tant le nombre de paramètres rentrant en ligne de compte est important. Chaque configuration géométrique est une source potentielle de raison de subdiviser un lien. De plus le nombre d'approximations effectuées au cours du calcul d'un échange radiatif limite fortement la crédibilité des résultats.

Les critères de raffinement existants sont valables dans leur configuration spécifique et aucun d'entre eux (sauf peut-être le critère BF) ne peut s'appliquer dans le cas d'une scène quelconque.

		Faible définition	Forté définition
Solution initiale :			
Temps de calcul	:	47 s.	931,4 s.
Nombre de liens	:	87.000	1.449.340
Mise à jour de la selection :			
Temps de calcul total	:	5 s.	39 s.
dont temps de raffinement	:	~ 5 s.	15 s.
Nombre de liens ajoutés	:	18.500	18.500

TAB. 4.5: Temps de mise à jour d'ombres à l'aide d'un raffineur collaboratif.

Les bornes de l'erreur commise dans un échange entre deux éléments sont très larges et divergent rapidement même dans le cas de visibilité totale en tout point.

L'intégration de la visibilité au sein d'un oracle de raffinement est le point le plus délicat, car on ne peut pas poser d'hypothèses quant aux positionnements des éléments de la scène. Toutes les configurations géométriques sont possibles et dans des scènes provenant de modeleurs quelconques, les pires configurations se présentent effectivement.

Le comportement global d'un oracle de raffinement est difficilement saisissable, car des milliers d'interactions lumineuses interviennent au cours du processus de raffinement, l'ordre de traitement des liens est important, les approximations faites sont difficilement prévisibles dû au sous-échantillonnage fréquent de la fonction de radiosit , ...

Il y a donc besoin d'une structure stable et g n rale du processus de raffinement pour pouvoir l' tudier. Il faut aussi un outil performant permettant de regarder minutieusement le comportement de l'oracle dans certaines configurations probl matiques.

Nous avons propos  une structure de raffinement bas e sur une d coupe du probl me en t ches  l mentaires : les *outils de raffinement*. Chaque outil est sp cialis  dans le traitement d'une seule information. Nous d finissons un outil comme une entit  capable de r pondre   une question simple.

Dans notre structure l'oracle pose une s rie de questions aux outils et, en fonction des r ponses obtenues, va guider plus ou moins sa d cision vers la d cision finale qu'il doit prendre. Le processus de raffinement peut  tre alors vu comme une discussion entre l'oracle et un ensemble d'outil. L'ensemble des discussions possible est repr sent  par un graphe orient  sans cycles ou les noeuds sont des outils et les feuilles les d cisions de l'oracle.

Nous avons pr sent  cette structure en d tail, mettant en avant les nombreux avantages intrins ques des graphes. Puis nous avons d taill  un ensemble d'outils utilis  couramment dans la construction de nouveaux raffineurs. Enfin nous donnons une application   un cas concret, l'int gration d'un crit re de visibilité am lior , guid  par la g n ration de statistiques automatique de la structure.

L'usage de notre structure permet une compr hension plus ais e du processus de raffinement gr ce   un cadre souple et bien d fini et facilite ainsi la g n ration de nouveaux oracles de raffinement.

Visibilité

Tu ne vois ton ombre que lorsque tu tournes le dos au soleil.

Khalil Gibran

La visibilité est une problématique largement étudiée dans le domaine de l'imagerie numérique, que ce soit en synthèse d'image, en robotique ou en vision par ordinateur. C'est une tâche relativement lourde en temps, qu'elle soit effectuée au niveau logiciel (intersection géométrique de rayons, élimination de partie cachées, ...) ou au niveau de capteurs physiques (scanners de distances, Z-caméras, capteurs ultrasons ou infrarouges, ...). La visibilité est fréquemment le goulot d'étranglement des algorithmes qui doivent l'évaluer.

Le terme générique "visibilité" regroupe un nombre de domaines relativement vaste. Certaines applications veulent connaître la visibilité à partir d'un unique point de vue, fixe ou mobile. D'autres désirent estimer les zones d'ombres et de pénombre à partir de source lumineuses ponctuelles, surfaciques ou volumiques. Les calculs de visibilité peuvent être effectués à différents niveaux de précision en fonction de l'importance du calcul... De bon survols de l'état actuel des connaissances peuvent être trouvés [WW92, FvFH90, Dur99, Dur00].

L'algorithme de radiosité utilise une évaluation de la visibilité au coeur de son noyau, lors du calcul du facteur de forme (cf. §1.2, page 27). Les configurations géométriques rencontrées sont très diverses et on ne peut pas utiliser d'a priori afin de simplifier les algorithmes. Nous nous intéresserons donc dans ce chapitre aux algorithmes de visibilité génériques entre deux éléments quelconques que nous appellerons *astres*, par analogie avec l'astronomie qui s'occupe de l'occultation entre les astres.

Nous commencerons par définir notre classification de la visibilité par une extension de la définition classique. Puis nous étudierons rapidement les différents algorithmes d'évaluation de la visibilité

utiles en radiosité hiérarchique, en étendant l'outil d'évaluation volumique et en optimisant l'outil ponctuel. Nous finirons par une étude statistique de la classification de la visibilité par un pinceau traversant un espace d'obstacles. Les résultats théoriques développés seront validés expérimentalement.

1 Classification de visibilité.

La classification de visibilité est différente de l'estimation du facteur de visibilité. Dans le cadre du raffinement utilisant les outils tels que nous les avons définis (cf. §2, page 71), les classes sont les réponses des outils effectuant cette classification de visibilité.

1.1 Classes.

Classes certaines.

Nous avons défini une classification suivant les différentes configurations de visibilité entre deux astres. Les classes de bases utilisées implicitement ou explicitement par n'importe quel logiciel qui doit utiliser de la visibilité sont :

- **visibilité totale** (VISIBLE) : il n'existe pas un seul segment reliant deux points quelconques des astres qui intersecte un obstacle ;
- **visibilité nulle** (INVISIBLE) : il n'existe pas un seul segment reliant deux points quelconques des astres qui n'intersecte pas au moins un obstacle ;
- **visibilité partielle** (PARTIEL) : il existe au moins un segment qui intersecte un obstacle et un segment qui n'intersecte aucun obstacle.

Nous appelons ces classifications *classifications certaines*.

Classes probables.

Nous avons étendu la classification certaine afin de prendre en compte la validité de la classification. Par exemple, si l'on utilise un échantillonnage des deux astres et une évaluation de la visibilité entre chaque échantillon, même si aucun segment n'est occulté on ne peut pas garantir qu'il n'existe pas un "petit obstacle raté" entre les astres. Il faut donc différencier la classification donnée par cette évaluation et la classification VISIBLE.

Il serait intéressant d'avoir un pourcentage de fiabilité de la réponse. Dans l'exemple précédent, connaître le plus petit obstacle qui ne peut pas être raté par échantillonnage, ou un facteur décrivant le volume représenté par chaque échantillon. Hélas, il est difficile de définir un critère commun calculable par tous les algorithmes d'évaluation de la visibilité.

Nous avons donc choisi une classification intermédiaire qui ne caractérise pas quantitativement mais uniquement qualitativement l'erreur commise. La classification dans notre exemple serait PROBABLEMENT VISIBLE. Trois nouvelles *classifications probables* sont ajoutées PROBABLEMENT VISIBLE, PROBABLEMENT INVISIBLE et PROBABLEMENT PARTIEL, correspondantes aux réponses certaines, mais dont le pourcentage de certitude n'est pas 100%.

Classes propres.

Quelques classes de visibilité peuvent être encore définies pour chaque algorithme de classification afin de donner des précisions sur la classification. Par exemple, une classification partielle peut être due à un groupe de polygones intersectant le volume contenu entre deux astres ou à une surface

qui le chevauche. Cette information peut-être utilisée, notamment dans le cas du raffinement, dans les algorithmes de radiosité.

En règle générale, le maximum d'information se doit d'être intégré dans la classification, quitte à multiplier le nombre de réponses possibles.

1.2 Volumes de vision.

Les *volumes de vision (shaft)* sont des outils qui permettent de déterminer si des obstacles se trouvent entre deux éléments. Contrairement à la technique de lancer de rayon, les volumes de vision sont volumiques et permettent de ne pas souffrir de phénomènes d'aliassage. Un volume de vision est un volume englobant l'ensemble des segments partant d'un astre vers le second : l'enveloppe convexe des deux astres (fig. 5.1a).

Il est difficile de créer cette enveloppe convexe pour toutes les configurations géométriques existantes, les cas d'intersections et d'interpénétrations donnant de nombreux cas particuliers en dimension 3. Les volumes de vision tels qu'ils sont actuellement utilisés sont l'enveloppe convexe des boîtes englobantes alignées avec les axes des éléments (fig. 5.1b), volume beaucoup plus facile à construire.

Ce volume peut être défini comme l'intersection finie de demi espaces, car il est par définition un polyèdre convexe. Haines et Wallace les représentent ainsi comme un ensemble de plans, frontières des demi espaces, et une boîte globale alignée avec les axes contenant au plus près les éléments (fig. 5.1c) [HW91, HW94].

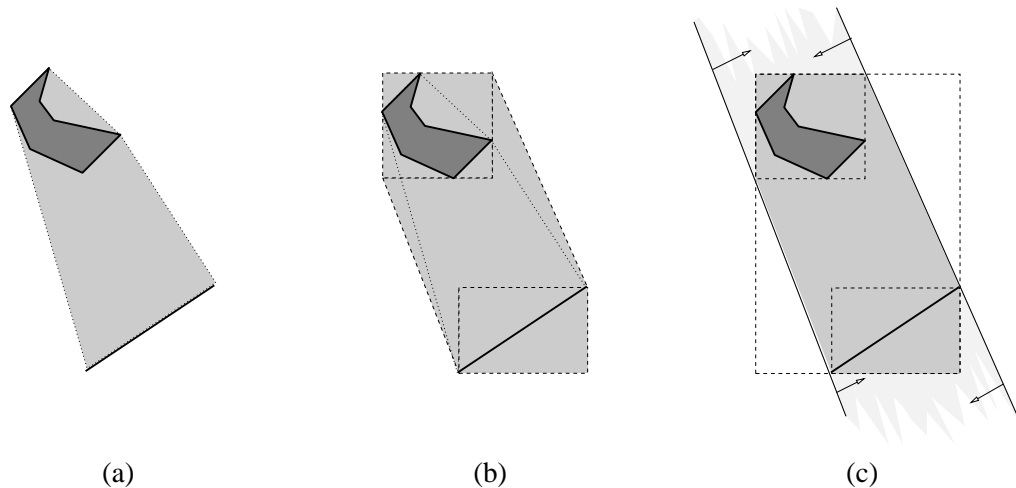


FIG. 5.1: Enveloppe convexe et volume de vision entre deux éléments. (a) Enveloppe convexe. (b) Volume de vision appuyé sur des boîtes englobantes alignées avec les axes. (c) Frontières des demi-espaces et boîte englobante au plus près.

Génération optimisée.

Nous nous plaçons dans le cadre de construction de volumes de vision en 3 dimensions entre deux boîtes englobantes alignées avec les axes que nous nommerons \mathcal{A} et \mathcal{B} (toutes les parties des algorithmes suivants restent notamment vraies en 2 dimensions)

Nous utilisons la création définie dans la seconde parution optimisée des volumes de vision [Hai00] que nous décrivons ici précisément afin de mettre en avant les différences lorsque nous décrivons les volumes de visions étendus (cf. §1.3, page 103).

L'optimisation, par rapport à l'algorithme initial, tire parti du fait que les scènes architecturales sont fréquemment composées de beaucoup d'objets alignés, les objets étant posés sur le sol ou sur d'autres objets, ceci dans des pièces souvent rectangulaires. Les volumes de vision engendrés peuvent fréquemment posséder une face perpendiculaire aux axes, dont la présence est inutile (fig. 5.2) car l'information de cette face est redondante avec celle de la boîte englobante globale.

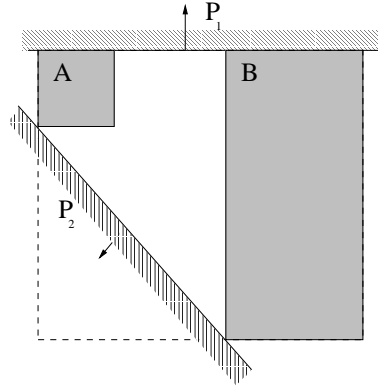


FIG. 5.2: Faces Inutiles dans un volumes de vision. Ici P_1 est redondante avec la boîte englobante.

Dans l'implémentation que nous proposons, nous utilisons deux tables de 6 bits relatives à chaque astre ($HasMinMaxA$ et $HasMinMaxB$). Chaque bit contient l'information que la boîte englobante des astres est, ou non, collée à la boîte englobante globale suivant les axes, ceci pour les minimums puis pour les maximums.

Le nombre de faces du volume de vision est de 8 dans le pire des cas. Elles sont déterminées par les couples d'arêtes correspondant au même axe et dont une et une seule est collée à la boîte globale. En utilisant les tables $HasMinMax$, ceci correspond à trouver certains couples (01, 10) et (10, 01) qui correspondent à une arête particulière des boîtes englobantes. Une des valeurs des tables étant forcément non nulle pour un indice donné, le test peut s'exprimer plus simplement en trouvant les couples (0?, ?0) et (?0, 0?) :

$$\left\{ (i, j) \in [0..5]^2 \text{ tels que } i > j, i \not\equiv j[3], \begin{array}{c} (\overline{HasMinMaxA[i]} \text{ et } \overline{HasMinMaxB[j]}) \\ \text{ou} \\ (\overline{HasMinMaxA[j]} \text{ et } \overline{HasMinMaxB[i]}) \end{array} \right\}^*$$

Dans l'exemple suivant (fig. 5.3), la boîte \mathcal{A} est collée à la boîte englobante pour la valeur minimale des trois axes mais pour aucune des valeurs maximales : soit la valeur de la table $HasMinMaxA$ de 111000[†]. Les couples d'indices décrivant les arêtes à relier sont (0, 4) $\equiv (x_{min}, y_{max})$ et (0, 5) $\equiv (x_{min}, z_{max})$.

* $i \not\equiv j[3]$: i n'est pas congru à j modulo 3

† 1 et 0 représentent respectivement les valeurs *vrai* et *faux*.

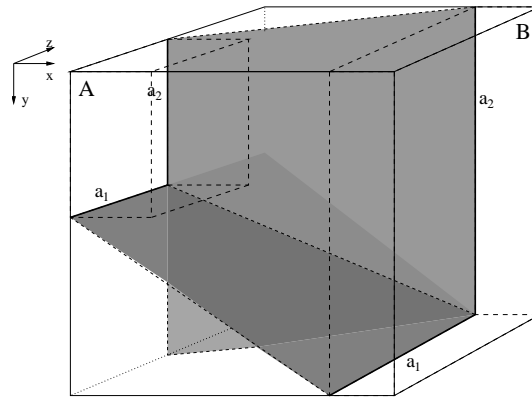


FIG. 5.3: Exemple de création d'un volume de vision en 3 dimensions.

axes	x_{min}	y_{min}	z_{min}	x_{max}	y_{max}	z_{max}
	0	1	2	3	4	5
<i>HasMinMaxA</i>	1	1	1	0	0	0
<i>HasMinMaxB</i>	0	1	1	1	1	1
couples	0				4	(axe a_1)
	0					5 (axe a_2)

Une fois les faces déterminées (algo. 5.1), il faut les créer. Les arêtes étant alignées avec les axes, leur expression est simple (algo. 5.2).

Intersections et classifications.

Les tests d'intersections avec des obstacles se font avec leurs boîtes englobantes de façon à accélérer les calculs et simplifier les cas à traiter. Pour un obstacle donné, ses positions possibles par rapport au volume sont EXTERNE, INTERNE et SUPERPOSÉ quand sa boîte englobante est respectivement complètement à l'extérieur du volume, complètement à l'intérieur, où qu'elle le chevauche.

Les deux catégories INTERNE et EXTERNE sont sûres à 100%. Par contre, l'utilisation de volumes englobants pour créer le volume de vision, fait que la réponse SUPERPOSÉ ne reflète pas obligatoirement la configuration exacte de l'obstacle par rapport aux astres. Dans la figure suivante (fig. 5.4), les obstacles A et B sont extérieurs au volume pour deux raisons différentes : A est extérieur à l'englobant des deux éléments, B est extérieur au volume en lui-même. L'obstacle C est interne au volume car sa boîte englobante est entièrement à l'intérieur du volume. Les configurations D et E sont erronées car les boîtes englobantes chevauchent effectivement le volume mais pas l'obstacle en lui-même.

Les différentes classifications de la visibilité sont liées à l'ensemble des réponses données pour l'ensemble des différents obstacles potentiels. Trois réponses sont possibles :

- VISIBLE : tous les obstacles potentiels sont EXTERNE au volume de vision ;
- PARTIEL : tous les obstacles potentiels sont INTERNE et/ou EXTERNE au volume de vision ;
- PROBABLEMENT PARTIEL : il existe au moins un obstacle qui est SUPERPOSÉ au volume de vision.

La dernière classification est immuable car il suffit qu'un seul obstacle chevauche le volume de vision pour que la réponse ne puisse plus être une autre. Ceci à l'avantage de limiter le temps d'exécution de l'outil car une fois cette réponse acquise, il n'est plus nécessaire de continuer à tester les suivants.

Algorithme 5.1 Génération optimisée de volumes de vision*Génération optimisée* sur \mathcal{A}, \mathcal{B} \mathcal{A}, \mathcal{B} : deux boîtes englobantes alignées avec les axes.1 - Remplissage des deux tables *HasMinMax***pour** n de 0 à 2 **faire** $HasMinMaxA[n] \Leftarrow \mathcal{A}[n] \leq \mathcal{B}[n]$ $HasMinMaxB[n] \Leftarrow \mathcal{B}[n] \leq \mathcal{A}[n]$ **fin pour****pour** n de 2 à 5 **faire** $HasMinMaxA[n] \Leftarrow \mathcal{A}[n] \geq \mathcal{B}[n]$ $HasMinMaxB[n] \Leftarrow \mathcal{B}[n] \geq \mathcal{A}[n]$ **fin pour**

2 - Création des plans utiles

pour i de 0 à 5 **faire** **pour** j de $i + 1$ à 5 **faire** **si** ($i+3 \neq j$ et ($(!HasMinMaxA[i]$ et $!HasMinMaxB[j])$ ou
 ($!HasMinMaxB[i]$ et $!HasMinMaxA[j])$)) **alors** **CreePlan**($\mathcal{A}, \mathcal{B}, i, j$) *Crée le plan supporté par les arêtes (i,j)* **fin si** **fin pour****fin pour****Algorithme 5.2** Création des plans à partir des arêtes*CreePlan* sur $\mathcal{A}, \mathcal{B}, i, j$ \mathcal{A}, \mathcal{B} : deux boîtes englobantes alignées avec les axes. i, j : les indices des directions des axes supports.*orientation d'un trièdre sur les axes par le triplet (a,b,c)* $a \Leftarrow i \bmod 3, \quad b \Leftarrow j \bmod 3, \quad c \Leftarrow 3 - a - b$ *calcul de la normale au plan (\vec{n}) et de sa distance à l'origine (d)* $d_u \Leftarrow \mathcal{B}[i] - \mathcal{A}[i], \quad d_v \Leftarrow \mathcal{B}[j] - \mathcal{A}[j]$ $\vec{n}[a] \Leftarrow d_v, \quad \vec{n}[b] \Leftarrow -d_u, \quad \vec{n}[c] \Leftarrow 0$ $d \Leftarrow \mathcal{B}[i] \cdot \mathcal{A}[j] - \mathcal{A}[i] \cdot \mathcal{B}[j]$ *orientation du plan vers l'extérieur du volume de vision***si** ($(d_u < 0$ ou $b \neq j)$ et $(d_u > 0$ ou $b = j)$) **alors** $\vec{n} \Leftarrow -\vec{n}, \quad d \Leftarrow -d$ **fin si****Hiérarchie de volumes de vision.**

Les volumes de vision peuvent être subdivisés en plusieurs sous-volumes ceci de manière hiérarchique. Dans l'application spécifique de la radiosité un volume de vision peut-être associé à un et un seul élément hiérarchique (de la hiérarchie de liens), créant une bijection entre la hiérarchie de volumes de vision et la hiérarchie de la solution de radiosité. Ceci permet dans le cadre de la radiosité

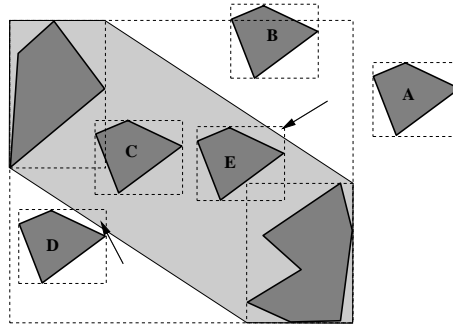


FIG. 5.4: Différentes configurations géométriques de classification de visibilité par volume de vision.

dynamique de trouver efficacement les liens qui doivent être mis à jour lors du déplacement d'un élément de la scène [DS97].

Les volumes de vision peuvent être modifiés pour garantir l'occultation totale (cf. §1.3, page 103). Ces volumes de visions étendus ont été utilisés en hiérarchie avec l'ajout d'un algorithme de regroupement de surface pour déterminer l'occultation totale due à de multiples obstacles jointifs [LP00]. Hélas les obstacles doivent être convexes pour garantir l'occultation totale. L'extraction de larges obstacles se limite alors à des surfaces quasi-planes organisées ou non en une hiérarchie propre [WHG99].

Volumes de vision orientés.

Les volumes de vision sont construits à partir des boîtes englobantes alignées avec les axes des éléments. Ceci pour une raison de facilité et de temps de calcul de création du volume, mais aux dépends d'un volume parfois fortement surévalué par rapport au volume défini par l'enveloppe convexe. Si l'on s'autorise un temps de création plus élevé, il est possible de créer des volumes de vision orientés beaucoup plus précis.

Certaines scènes où les objets sont allongés et/ou décorrélés des axes, donnent de meilleurs résultats avec l'utilisation de volumes de vision orientés [DPC01].

1.3 Volumes de vision étendus.

Les *volumes de visions étendus* sont des volumes de vision qui permettent de garantir l'occultation totale des deux astres par un unique obstacle convexe. L'idée est de combiner l'évaluation volumique des volumes de vision avec une évaluation par échantillonnage sur ses arêtes : les *arêtes longitudinales*. Si toutes les arêtes longitudinales sont occultées et que l'obstacle est convexe, alors la visibilité est nulle entre les astres.

Nous allons présenter un algorithme de génération des volumes de vision étendus optimisé dans le cas de boîtes englobantes alignées avec les axes, puis les différentes réponses de classification de la visibilité ainsi que la façon de les obtenir en utilisant une hiérarchie de boîtes englobantes.

Génération optimisée.

Nous reprenons la génération de volume de vision précédemment décrite (cf. §1.2, page 99), à laquelle nous ajoutons la détermination des arêtes longitudinales avec un minimum de calculs. Le nombre de faces du volume de vision est de 8 dans le pire des cas, c'est aussi le nombre maximal

d'arêtes car le volume est fermé. Par contre le nombre d'arêtes n'est généralement pas égal au nombre de faces car les faces inutiles ne sont pas stockées (fig. 5.2). Toutes les arêtes longitudinales doivent l'être.

Une face est porteuse de deux arêtes longitudinales aisément définies par les points extrémités. Nous utilisons une table temporaire des arêtes déjà créées afin de ne pas dupliquer leur création. Les sommets sont numérotés de 0 à 7 codés sur 3 bits où chaque bit détermine si le sommet est au minimum (valeur 0) ou au maximum (valeur 1) de la boîte englobante générale, ceci sur les trois axes. Les arêtes longitudinales relient deux sommets positionnés de même indice, il suffit donc de stocker les numéros des sommets pour connaître l'arête à créer.

Lorsque l'on a déterminé une paire d'arêtes à relier par un couple (i, j) , il faut obtenir les numéros des points à relier. Ceci s'obtient par la formulation pseudocode suivante :

Algorithme 5.3 Calcul des numéros des sommets de l'arête déterminée par le couple (i, j)

le numéro du point dépend de la position – minimale ou maximale – dans la boîte englobante

$$n1 \leftarrow (i >= 3) \cdot 2^{(i \bmod 3)} + (j >= 3) \cdot 2^{(j \bmod 3)}$$

$$n2 \leftarrow n1 + 2^{(3 - (i \bmod 3) - (j \bmod 3))}$$

où mod est l'opérateur de reste de la division euclidienne.

Intersections et classifications.

La classification de visibilité des volumes de vision étendus doit être légèrement modifiée par rapport à celle des volumes de vision initialement décrits. La classification immuable n'est plus PROBABLEMENT PARTIEL mais INVISIBLE. Ce qui veut dire que l'on doit tester tous les éléments jusqu'à l'obtention d'une invisibilité totale. Ce cas étant plus ou moins fréquent en fonction de la scène utilisée, il convient d'utiliser cet outil plus lourd en temps de calcul à bon escient. D'où la motivation de notre étude de visibilité (cf. §4, page 113).

L'occultation totale n'est évaluée que sur les polygones. Elle dépend de l'intersection de chaque arête longitudinale avec celui-ci. Les polygones à tester peuvent se limiter à ceux qui ont une aire projetée supérieure à celle de la section du volume de vision au niveau de leur intersection, ce qui limite le coût de l'évaluation de la visibilité nulle.

Une solution serait de trier les polygones selon leur aire. Ainsi, les obstacles potentiels ayant une aire inférieure à celle de la section minimale du volume de vision ne seraient plus utiles à la détermination de la classification INVISIBLE. Le tri ne peut pas être précalculé car il dépend de l'orientation du pinceau dans les groupes de surfaces obstacles, ce qui réduit fortement l'intérêt de cette méthode.

2 Techniques d'accélération.

Les différentes techniques de classification de la visibilité et d'évaluation du facteur de visibilité sont relativement lourdes en temps de calculs. Nous présentons dans ce chapitre une étude pratique du temps utilisé par l'évaluation de la visibilité dans le cadre de la radiosité hiérarchique. Nous présenterons aussi une optimisation pour chacun des deux principaux algorithmes traitant la visibilité.

2.1 Motivation.

Dans un algorithme de radiosité il est généralement admis que le temps utilisé à l'évaluation de la visibilité comme le plus important. Des chiffres de 50 à 80% ont été avancés dans différents articles et thèses.

Algorithme 5.4 Génération optimisée de volumes de vision étendus**Génération optimisée** sur \mathcal{A} , \mathcal{B} \mathcal{A} , \mathcal{B} : deux boîtes englobantes alignées avec les axes.1 - Remplissage des deux tables *HasMinMax*

identique à l'algorithme de génération de volumes de vision normal (algo. 5.1)

2 - Création des plans utiles et pointage des arêtes

pour i de 0 à 5 **faire** **pour** j de $i + 1$ à 5 **faire** *Cas des plans effectifs* **si** ($i+3 \neq j$ et ($(\neg \text{HasMinMaxA}[i]$ et $\neg \text{HasMinMaxB}[j])$ ou
 ($\neg \text{HasMinMaxB}[i]$ et $\neg \text{HasMinMaxA}[j])$)) **alors** | **CréePlan**($\mathcal{A}, \mathcal{B}, i, j$) *Crée le plan (i,j)* | **PointeSommets**(i, j) *Pointe les deux arêtes du plan (i,j)* **fin si** *Cas des plans inutiles – Pointe uniquement les deux arêtes* **si** ($i+3 \neq j$ et ($\text{HasMinMaxA}[i]$ et $\text{HasMinMaxB}[j]$ et
 $\text{HasMinMaxB}[i]$ et $\text{HasMinMaxA}[j]$)) **alors** | **PointeSommets**(i, j) *Pointe les deux sommets extrémités)* **fin si** **fin pour****fin pour**

3 - Création des arêtes pointées

pour n de 0 à 7 **faire** *Crée uniquement les arêtes précédemment pointées* **si** (**EstPointe**(n)) **alors** | **CréeArete**($\mathcal{A}, \mathcal{B}, n$) **fin si****fin pour****CréeArete** sur \mathcal{A} , \mathcal{B} , i , j \mathcal{A} , \mathcal{B} : deux boîtes englobantes alignées avec les axes. n : l'indice de la paire de points à lier. $P_A[0] \Leftarrow (n \& 1 > 0) ? \mathcal{A}[0] : \mathcal{A}[3]$ $P_B[0] \Leftarrow (n \& 1 > 0) ? \mathcal{B}[0] : \mathcal{B}[3]$ $P_A[1] \Leftarrow (n \& 2 > 0) ? \mathcal{A}[1] : \mathcal{A}[4]$ $P_B[1] \Leftarrow (n \& 2 > 0) ? \mathcal{B}[1] : \mathcal{B}[4]$ $P_A[2] \Leftarrow (n \& 4 > 0) ? \mathcal{A}[2] : \mathcal{A}[5]$ $P_B[2] \Leftarrow (n \& 4 > 0) ? \mathcal{B}[2] : \mathcal{B}[5]$ où $\&$ est l'opérateur *et logique* bit à bit et $?$: l'opérateur ternaire d'exécution conditionnelle

Grâce à l'utilisation des statistiques fournies par la structure de graphe de raffinement, nous proposons une étude des temps utilisés par les outils de visibilité dans un graphe particulier.

Les raffineurs testés sont : le critère BF dont la visibilité est testée avec 4 échantillons et un second basé sur un critère BF dont la visibilité est classifiée par un outil utilisant les volumes de vision étendus pour des pinceau de section importante et par un échantillonnage pour des sections plus faibles.

La scène de test est constituée de deux bureaux collés séparés par une porte où une seule des deux pièces est éclairée. Le nombre initial de polygones avant raffinement est de 13.523 avec de nombreux petits objets ainsi que de grands obstacles comme le mur central (fig. 5.5).

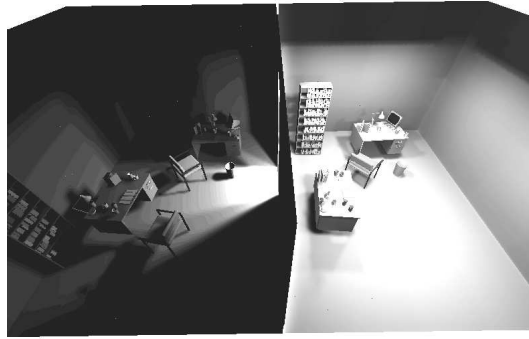


FIG. 5.5: Scène de test des tests de visibilité composée de multiples petits objets ainsi que d'un grand obstacle central.

	1 ^{ère} itération	2 ^{ème} itération
temps de la solution (s) :	42.7	126.8
nombre de liens total :	13857	83229
Outil de classification par échantillonnage (4 rayons)		
nombre d'appels :	73.400	145.000
temps total (%) :	64%	61%
temps par appel (ms) :	0.376	0.519

TAB. 5.1: Temps de raffinement utilisé par les outils de visibilité pour un raffineur basé sur un critère BF dont la visibilité est évaluée par échantillonnage.

Le temps imparti à la visibilité dépend beaucoup de l'algorithme utilisé ainsi que de la précision requise par l'utilisateur. Dans les deux exemples de raffineurs que nous avons utilisé, le pourcentage d'utilisation du temps va de 60% à presque 90% dans le cas de l'utilisation de volumes de vision étendus. De plus, dans notre mise en œuvre, les intersections de rayons au travers de la scène sont optimisés par l'utilisation d'un cache de rayons.

Le temps d'évaluation de la classification de visibilité par un volume de vision étendu est de l'ordre de 3 fois supérieur à une classification effectuée par échantillonnage de 4 rayons. Mais ce temps dépend aussi de l'environnement et du nombre de réponses INVISIBLE par évaluation du volume de vision. Dans cette scène, le nombre de classifications INVISIBLE est seulement de 7.2% et 3.5% respectivement pour la 1^{ère} et la 2^{ème} itération.

Quel que soit l'algorithme de visibilité utilisé on peut considérer que le temps passé à l'évaluation de la visibilité est supérieur à la moitié du temps de raffinement et peut monter à presque la totalité dans des configurations défavorables. L'optimisation de la visibilité aura donc un impact très important sur l'optimisation générale de l'algorithme de radiosité.

C'est pourquoi nous nous sommes particulièrement penchés sur ce problème.

2.2 Lancer de rayons.

Le test d'intersection avec un rayon peut-être de différentes natures. On peut chercher à savoir si le rayon est occulté ou non (réponse binaire). On peut chercher à connaître l'ensemble des obstacles

	1 ^{ère} itération	2 ^{ème} itération
temps de la solution (s) :	91.7	303.0
nombre de liens total :	15022	73110
Outil de classification par échantillonnage (4 rayons) :		
nombre d'appel :	16.700	12.250
temps total (%) :	8%	1.5%
temps par appel (ms) :	0.444	0.416
Outil de classification par volume de vision étendus :		
nombre d'appel :	56.900	137.400
temps total (%) :	76%	85%
temps par appel (ms) :	1.224	1.892

TAB. 5.2: Temps de raffinement utilisé par les outils de visibilité pour un raffineur basé sur un critère BF dont la visibilité est évaluée par échantillonnage et par volumes de vision étendus.

qui occultent le rayon. On peut aussi vouloir connaître la ou les distances minimales du rayon à tous les objets.

Optimisations existantes.

Le lancer de rayon à fait l'objet de nombreuses optimisations. Les plus utilisées pour leur facilité d'implémentation et les gains obtenus sont :

- Cache : on teste l'intersection avec les derniers obstacles précédemment intersectés. Cette optimisation n'est utile que quand on désire avoir une réponse binaire d'occultation du rayon. Les résultats dépendent de la cohérence spatiale de la scène aux alentours du rayon. La fonction d'intersection retourne *vrai* ou *faux* selon que le rayon est occulté ou non.
- Hiérarchie : en générant une hiérarchie de volumes englobants contenant la géométrie de la scène, on peut éliminer directement les éléments dont le volume englobant n'est pas intersecté par le rayon (fig. 5.6a). Les tests d'intersections se font uniquement au niveau des feuilles (les surfaces). Il est intéressant d'utiliser des volumes englobants dont l'intersection est rapidement calculable, soit des sphères et des pavés alignés avec les axes pour les plus simples. On peut utiliser la hiérarchie dans tous les cas, y compris en complément d'un cache de rayons. Par contre elle est assez coûteuse en mémoire. La fonction d'intersection retourne l'objet intersecté.

Parcours feuille-feuille.

En radiosité hiérarchique, nous avons déjà une hiérarchie de la scène utilisable pour optimiser les intersections de rayons. Il est fréquent de faire plusieurs tests d'intersections de rayons entre les deux mêmes éléments hiérarchiques pour calculer le facteur de visibilité entre eux. Dans un processus de raffinement classique, les éléments traités successivement sont très proches. Le cache de rayons est donc intéressant, car la cohérence spatiale est forte.

Si un rayon n'est pas intersecté, la hiérarchie est parcourue depuis la racine jusqu'aux obstacles potentiels. Ceux-ci sont très probablement géométriquement proches des derniers objets intersectés.

Notre idée est de tirer parti de la cohérence spatiale des éléments traités et de la cohérence spatiale des obstacles. Pour cela nous allons modifier le parcours de la hiérarchie. Du fait de la cohérence des éléments traités, nous voulons tester en premier lieu le dernier obstacle intersecté, tout comme le

fait un cache de rayons. S'il n'est pas obstacle, nous allons tester ses voisins géométriques proches, en parcourant la hiérarchie à partir de celui-ci, technique que nous avons appelé *intersection feuille-feuille* (fig. 5.6b).

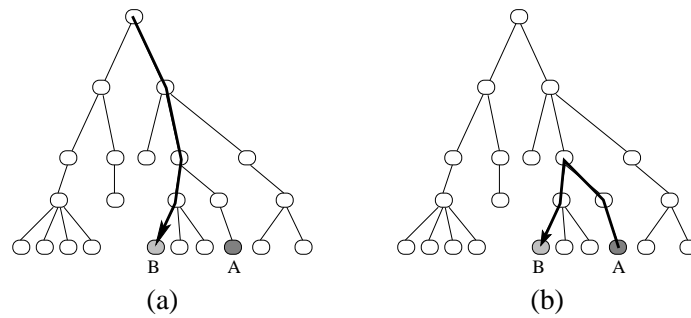


FIG. 5.6: Parcours hiérarchique (a) et feuille-feuille (b) pour le calcul d'intersection de rayon (réponse binaire). A : dernier obstacle, B : obstacle actuel.

Nous devons donc connaître le père de l'obstacle potentiel, et toute son ascendance dans le pire des cas. Soit l'on utilise une hiérarchie doublement chaînée où chaque élément connaît son père, soit une hiérarchie simplement chaînée. Nous allons traiter ce dernier cas car il est plus général et parce que le coût mémoire engendré par un chaînage double est important en comparaison du petit nombre d'informations facilement accessibles que nous allons retrouver.

Nous avons besoin de connaître la branche complète de la hiérarchie qui relie le dernier objet intersecté à la racine afin de pouvoir la remonter rapidement. Nous avons toute cette information pendant le calcul d'intersection et il suffit de modifier la valeur de retour de la fonction d'intersection afin de la sauvegarder.

Dans le pire des cas, le nombre de calculs d'intersections fait est plus élevé dans le cas du parcours inverse (feuille-feuille) que dans le parcours classique. Il rajoute en effet la remontée de la branche avec calcul d'appartenance à chaque noeud. Mais dans le cas moyen, nous pouvons espérer que la cohérence soit suffisante. La différence de temps entre les deux méthodes est fonction de la cohérence des obstacles dans le processus de raffinement et de la profondeur de la hiérarchie.

Nous allons faire une comparaison entre l'optimisation par cache et l'optimisation par parcours feuille-feuille pour en mesurer l'intérêt. Les caches de rayons que nous utilisons possèdent une taille fixe, nous avons testé des tailles de cache de 1 à 10 rayons.

Résultats.

Nous avons utilisé une première scène de test bien adaptée à la morphologie de l'algorithme, c'est-à-dire une scène comportant une multitude de petits polygones géométriquement très proches afin que la cohérence des obstacles soit très élevée. Nous avons testé l'algorithme sur une seconde scène comportant de très grands obstacles, où l'algorithme d'intersection feuille-feuille doit être moins avantageux. Les scènes utilisées sont un albisia seul éclairé par le soleil (fig. 5.7a) et un labyrinthe éclairé par quelques sources lumineuses (fig. 5.7b).

Dans le cas de l'albisia, nous avons appliqué un fort raffinement afin de multiplier les requêtes de visibilité (presque 10 millions) (tab. 5.3). Le nombre de requêtes est beaucoup plus faible dans le cas du labyrinthe (de l'ordre du million) (tab. 5.4).

Nous utilisons le nombre d'intersections obtenu par un parcours de la hiérarchie sans cache comme référence. Les caches de rayons sont testés pour des tailles allant d'un cache simple (uniquement le dernier obstacle stocké) jusqu'à des caches de taille 10 (les 10 derniers obstacles stockés).

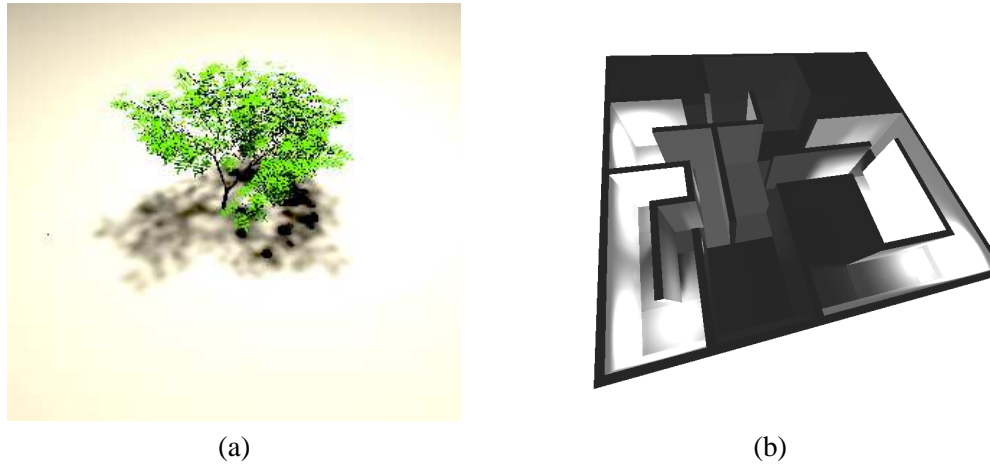


FIG. 5.7: Scènes de tests de l'algorithme de parcours inverse. (a) scène de l'*albisia* ayant une forte cohérence géométrique des obstacles. (b) scène du *labyrinthe* possédant peu d'obstacles de taille importante.

Les temps d'intersection d'une cellule (noeud de la hiérarchie de la scène) et d'un polygone (feuille) ne sont pas équivalents. Nous présentons le nombre d'intersections pour chacun de façon séparée afin de mettre en évidence la distance parcourue par chaque algorithme dans la hiérarchie de la scène.

Scène de l'*albisia* :

9.839.507 requêtes de visibilité

	intersections de surfaces (en millions)		intersections de cellules (en millions)	
Hiérarchie	: 218,0		535,7	
Feuille-feuille	: 197,4	(-9.4%)	394,2	(-26.4%)
Cache (1)	: 212,6	(-2.5%)	474,4	(-11.4%)
Cache (3)	: 218,8	(+0.4%)	434,2	(-18.9%)
Cache (5)	: 217,5	(-0.2%)	378,2	(-29.4%)
Cache (7)	: 225,9	(+3.6%)	361,7	(-32.5%)
Cache (10)	: 243,1	(+11.5%)	353,6	(-34.0%)

TAB. 5.3: Évaluation du nombre d'intersections calculées dans la scène de l'*albisia* en fonction du type d'intersections utilisé.

On remarque que le gain en nombre de calculs d'intersections de surfaces est effectivement légèrement supérieur à celui d'un cache d'intersections car les obstacles géométriquement proches sont aussi proches au sein de la hiérarchie. La recherche d'obstacles dans la hiérarchie à partir du dernier élément intersecté permet de les trouver plus rapidement en testant moins de surfaces.

Scène du labyrinthe :

753.116 requêtes de visibilité

	intersections de surfaces (en millions)		intersections de cellules (en millions)	
Hiérarchie	: 128,1		5,7	
Feuille-feuille	: 74,0	<i>(-42.2%)</i>	4,7	<i>(-17.4%)</i>
Cache (1)	: 74,5	<i>(-41.8%)</i>	4,5	<i>(-21.1%)</i>
Cache (3)	: 69,5	<i>(-45.8%)</i>	4,4	<i>(-22.9%)</i>
Cache (5)	: 69,9	<i>(-45.4%)</i>	4,4	<i>(-23.0%)</i>

TAB. 5.4: Évaluation du nombre d'intersections dans la scène du labyrinthe en fonction du type d'intersections utilisé.

Par contre le gain d'intersections de groupes de surfaces est plus faible. En effet, les remontées de la hiérarchie obligent un nombre d'intersections plus important de cellules que ne le ferait une simple descente.

Parmi les caches de rayons, celui stockant les 5 derniers obstacles est le meilleur compromis permettant de réduire au maximum le nombre d'intersections de cellules et de surfaces. Par rapport à ce cache, les gains du parcours feuille-feuille sont très faibles. L'implémentation étant plus difficile, l'intérêt de cette méthode est très limité.

3 Volumes de vision et raffinement.

Le raffinement en radiosité hiérarchique est un processus récursif où chaque lien peut-être subdivisé en d'autres liens contenus dans le volume reliant l'émetteur et le récepteur. Les volumes englobants des éléments subdivisés sont inclus dans le volume englobant de l'élément immédiatement supérieur et les volumes de vision fils sont donc inclus dans le volume de vision père (fig. 5.8). L'ensemble des obstacles potentiels déterminé par volumes de vision est, lui aussi, inclus dans l'ensemble des obstacles potentiels du père de l'élément traité.

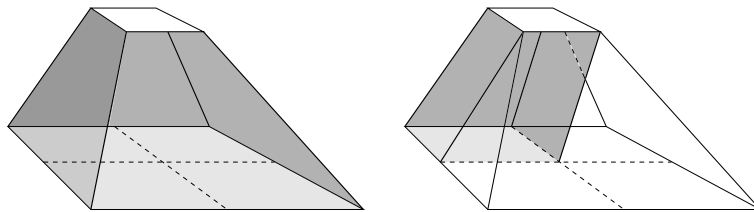


FIG. 5.8: Inclusion des volumes de visions entre un niveau hiérarchique de raffinement et les volumes inférieurs engendrés.

En connaissant l'ensemble des obstacles potentiels du père ou d'un ascendant quelconque lors du même processus de raffinement, on peut limiter le nombre de tests d'intersections du volume de

vision. Partant de cette idée, nous avons construit une liste d'optimisation de taille fixe définie par l'utilisateur.

Notre but est de limiter le nombre d'intersections calculées en réutilisant la liste des obstacles potentiels d'un ascendant, si elle existe. Ceci sera fait sans effectuer aucun test d'intersection supplémentaire. Au niveau mémoire, il est trop coûteux de sauvegarder cette liste pour chaque élément.

Il est à noter que cette optimisation ne peut fonctionner que dans le cadre d'un parcours en profondeur de l'arbre de raffinement par lien. Ce qui élimine les algorithmes scindant les processus de raffinements en processus indépendants, tel que certains algorithmes parallèles [SH00].

3.1 Liste de volumes de vision.

Le processus de raffinement d'un lien est un processus récursif de subdivision et le volume de vision d'un lien fils obtenu par raffinement est inclus dans le volume de vision de son père. Ainsi, les obstacles potentiels d'un volume de vision fils forment un sous-ensemble des obstacles potentiels de son père. En sauvegardant la liste des obstacles pour chaque niveau hiérarchique, les tests d'intersections des niveaux inférieurs pourront se réaliser avec cette liste.

Nous utilisons une table des obstacles potentiels, ainsi qu'une table de hachage (*hash-table*) permettant de retrouver rapidement dans la table des obstacles le début et la fin d'un niveau hiérarchique donné. La liste contient les obstacles des différents niveaux hiérarchiques dans l'ordre d'insertion durant le parcours récursif (ordre croissant). Chaque niveau représente une sauvegarde de l'état de test de la visibilité dans la hiérarchie de la scène pour un niveau donné.

L'ensemble des obstacles de toute la scène pour tous les niveaux hiérarchiques peut être très volumineux en mémoire. Pour éviter une saturation de l'espace mémoire, nous avons utilisé une table de taille fixe. Lorsque la table est pleine, le niveau courant n'est pas validé à la fin du processus de raffinement et ne sera donc pas utilisé ultérieurement.

Dans le cadre de la structure de raffinement proposée dans ce mémoire (cf. chapitre 4, page 69), la liste de volumes de vision est gérée par le gestionnaire de données. Les outils de classification de visibilité utilisant les volumes de vision doivent lui indiquer tous les obstacles testés avec leur position par rapport au volume, ainsi que tous les obstacles qui n'ont pas été testés après qu'une classification immuable ait été trouvée.

L'ensemble des outils de visibilité peut réutiliser les informations de la liste des obstacles potentiels, y compris les outils utilisant des rayons. Ceci permet d'augmenter la rapidité des intersections obstacles-rayons en éliminant, sans frais, les éléments qui ne peuvent pas être des obstacles potentiels.

Nous décrivons maintenant plus en détail la sauvegarde et la récupération des obstacles potentiels à l'aide de notre liste d'optimisation.

Sauvegarde des obstacles potentiels.

Avant chaque processus de raffinement d'un lien, c'est-à-dire à chaque fois que l'on se déplace à un nouveau noeud de la hiérarchie de raffinement, il faut indiquer à la liste le début d'un nouveau niveau.

La table de hachage place les curseurs de début et de fin de niveau juste après le premier niveau inférieur valide afin de pouvoir y placer les obstacles potentiels de ce niveau.

S'il y a utilisation d'un outil de volumes de vision, les obstacles potentiels sont insérés dans la liste après le dernier élément correspondant au niveau hiérarchique courant. Seuls les obstacles qui ne sont pas entièrement en dehors du volume de vision courant doivent être sauvegardés. Le marqueur de fin de niveau est incrémenté à chaque entrée.

A la fin du processus de raffinement d'un niveau donné, si la liste a été saturée, le niveau n'est pas validé en supprimant son entrée dans la table de hachage. Dans le cas contraire la liste est laissée dans l'état.

Lorsque l'on remonte d'un niveau dans la hiérarchie de raffinement, il suffit de supprimer l'entrée correspondant au niveau terminé de la table de hachage.

Récupération des obstacles potentiels.

Nous utilisons la convention que le niveau 0 de la liste d'optimisation est le niveau le plus faible, il ne contient exclusivement que la cellule racine de l'arbre de visibilité. Dans notre liste d'optimisation, n'importe quel niveau de la table inférieur un niveau donné, peut être utilisé comme liste des obstacles potentiels. Plus le niveau est élevé, plus le nombre d'obstacles potentiels est réduit. Le meilleur compromis est donc de trouver le niveau le plus élevé strictement inférieur au niveau courant.

Lorsque l'on désire récupérer la liste des obstacles potentiels pour un niveau de la hiérarchie de raffinement, il suffit donc de parcourir la table de hachage en remontant à partir du niveau courant jusqu'à trouver un niveau valide. Les obstacles potentiels sont compris entre les deux bornes définies dans la table de hachage.

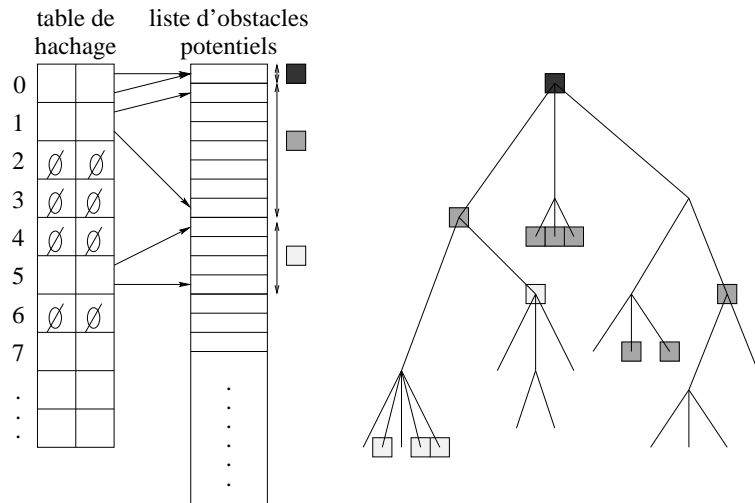


FIG. 5.9: Table de hachage et table des obstacles potentiels pour l'optimisation d'intersections avec des volumes de vision.

3.2 Résultats.

Nous avons utilisés la même scène de test que pour le calcul du temps de raffinement utilisé par la visibilité (fig. 5.5) en augmentant la précision de la solution afin de multiplier le nombre de classifications de visibilité et pouvoir faire des statistiques plus précises.

Nous présentons le nombre d'appels récursif de l'outil de discrimination de la visibilité par volumes de vision car le nombre total d'appels de l'outil est évidemment identique dans les deux cas. Par contre, à chaque appel de l'outil volume de vision, l'usage de la liste limite le nombre de descentes récursives dans l'arbre de visibilité.

Un niveau de la liste d'optimisation est un cliché de l'état de parcours de l'arbre de visibilité entre deux astres. Les liens fils peuvent commencer l'investigation d'un obstacle à partir d'un de l'état d'un de leur ascendant. (tab. 5.5).

	Sans liste d'optimisation	Avec liste d'optimisation
Temps de raffinement (s)	: 452	377 (-16.6%)
Nombre de liens	: 81.500	81.500
Nombre d'appel récursifs (millions)	: 49.34	35.69 (-27.66%)
Temps par appel (ms)	: 0.821	0.637 (-22.4%)

TAB. 5.5: Limitation des appels d'évaluation de la classification par volume de vision grâce à l'utilisation d'une sauvegarde du parcours du graphe de visibilité à un niveau précédent.

Sur cet exemple, le nombre d'appels récursifs a été réduit d'un quart grâce à l'utilisation de notre liste d'optimisation, ce qui a engendré une diminution de plus de 15% sur les temps de calculs.

La mise en œuvre n'est pas très compliquée et offre des résultats satisfaisants. De plus, la mémoire utilisée peut être limitée sans frais à une taille maximale de la liste d'optimisation. Dans notre exemple, nous avons utilisé une taille de table de 10.000 obstacles potentiels, limite qui n'a jamais été atteinte. Cette technique est, par contre, limitée à l'utilisation d'un parcours de l'arbre de raffinement en profondeur d'abord.

4 Statistiques de visibilité.

À l'heure actuelle, les recherches sur la visibilité tentent de trouver des simplifications, des cadres de travail, afin de pouvoir résoudre un problème de visibilité spécifique. Le problème est souvent simplifié par un ensemble d'hypothèses portant sur la position, l'orientation, etc. d'un ou des deux astres. Par exemple, lors d'un déplacement au sol dans un monde virtuel, certaines méthodes utilisent des cellules de visibilité discrètes, car la cohérence de la position de l'observateur est garantie.

Durant le processus de raffinement de la radiosité hiérarchique, la visibilité doit être évaluée de manière volumique, elle peut couvrir la totalité de la scène, la cohérence spatiale n'est pas garantie et toutes les paires d'astres de la scène peuvent être utilisées. Nous ne pouvons donc pas poser de contraintes quant aux astres. Les deux informations de classification et d'évaluation du facteur de visibilité sont importantes, toutefois, l'étape de classification précède celle d'évaluation et permet parfois de tirer une conclusion immédiatement, réduisant ainsi la charge de calculs.

Notre but est d'arriver à fournir une classification de la visibilité approximative à faible coût grâce à des outils géométriques.

4.1 Courbes de référence.

Le comportement de la visibilité dans sa globalité reste encore partiellement incompris. Dans un premier temps, nous allons chercher à décrire son comportement théorique en éliminant un certain nombre de paramètres.

Scènes de tests.

Dans un cube de taille définie, un ensemble d'obstacles est généré aléatoirement en fonction d'un paramètre de densité d'obstacles et de volume d'un obstacle. Le type d'obstacle que nous avons choisi est le cube car il est volumique et faible en description géométrique (fig. 5.10).

Dans tout le reste de cette section, nous allons utiliser les notations définies ci-dessous :

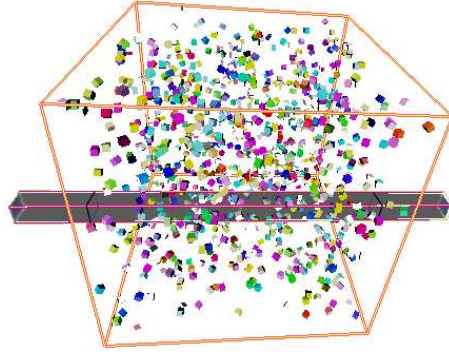


FIG. 5.10: Protocole de test utilisé permettant d'isoler les paramètres de taille et nombre d'obstacles, ainsi que la taille du pinceau (=taille des astres).

- \mathcal{V} le volume total du cube de test, S l'aire d'une face de ce cube et L la longueur d'une arête.
- v_0 le volume d'un obstacle, $r_0 = \sqrt[3]{\frac{3v_0}{4\pi}}$ le rayon de la sphère obstacle équivalente de volume v_0 et s_0 l'aire de la projection de cette sphère dans n'importe quelle direction.
- d la densité des obstacles à l'intérieur du cube global.
- $N = \frac{d \cdot \mathcal{V}}{v_0}$ le nombre d'obstacles (entier) dans le cube global.
- S_p l'aire de la section du pinceau et R_p le rayon de la section circulaire équivalente d'un pinceau de section S_p .

Les tests d'intersections se font sur un ensemble de pinceaux entre deux astres de taille variable tirés aléatoirement sur les bords du volume d'obstacles. Les pinceaux sont parallèles à une arête du cube d'obstacles. Dans nos tests pratiques, le cube global mesure 100 unités (u) de long, soit un volume de $10^6 u^3$. Nous avons effectué un échantillonnage de la section du pinceau (axe des abscisses des courbes) sur 100 valeurs. Pour chaque section, 5000 tests de classification sur des pinceaux aléatoires sont exécutés afin de pouvoir faire des statistiques fiables.

Les temps de calculs sont très longs, surtout pour calculer la solution de référence. Celle-ci utilise un échantillonnage brutal où la moitié des échantillons sont placés sur les arêtes et l'autre moitié est répartie uniformément à l'intérieur de l'élément. Tous les échantillons sont reliés entre eux. Le minimum que nous avons fixé est de 32 échantillons (soit 1024 rayons), le maximum est de 256 (soit 65536 rayons). Le maximum est relativement faible compte tenu de la taille élevée des dernières sections mais les temps de calculs nous limitent fortement.

Nous avons pris comme hypothèse que les obstacles sont positionnés entièrement aléatoirement selon une loi normale. Ils peuvent donc se chevaucher et la somme des volumes des obstacles peut dépasser le volume total du cube global. Ceci nous permet d'obtenir des événements indépendants et de combiner les probabilités de chaque obstacle.

Probabilités combinées.

Quel que soit le pinceau intersectant le volume, on peut déterminer la probabilité de classification de la visibilité en fonction des obstacles. Pour un seul obstacle donné, le pinceau a une probabilité p_{ext} que l'obstacle soit complètement à l'extérieur du pinceau et une probabilité p_{occ} que l'obstacle occulte complètement le pinceau. Pour un seul obstacle, ces probabilités sont équivalentes aux probabilités de classification pour un seul obstacle.

Quand il y a N obstacles, nous devons combiner les probabilités. La probabilité qu'un pinceau soit complètement visible ($N p_{vis}$) est la probabilité que tous les obstacles soient extérieurs. La probabi-

lité qu'un pinceau soit complètement occulté (${}^N P_{inv}$) est la probabilité complémentaire à ce qu'aucun obstacle n'occulte le pinceau. La classification étant ternaire – VISIBLE, INVISIBLE ou PARTIEL – la probabilité qu'un pinceau soit partiellement occulté (${}^N P_{par}$) est la complétion des deux précédentes probabilités à 1 (eq. 5.1).

$$\begin{aligned} {}^N P_{vis} &= ({}^I P_{ext})^N \\ {}^N P_{inv} &= 1 - (1 - {}^I P_{occ})^N \\ {}^N P_{par} &= 1 - {}^N P_{inv} - {}^N P_{vis} \end{aligned} \quad (5.1)$$

Simplification de l'étude.

L'ensemble des paramètres qui influent sur la visibilité est relativement vaste. Les deux astres tout comme l'ensemble des obstacles interviennent, voici la liste des paramètres que nous avons discriminés comme influants :

- Taille des astres.
- Distance entre les astres.
- Forme de la section.
- Homogénéité de la taille de la section entre les astres.
- Forme des obstacles.
- Nombre et taille des obstacles.
- Position des obstacles par rapport aux astres.

Nous ne prenons pas en compte l'occultation totale par plusieurs obstacles dont les projections se chevauchent entièrement sur le pinceau. Malgré nos recherches, nous n'avons pas trouvé d'estimateur valable d'une telle occultation combinée.

4.2 Pinceau infinitésimal.

La probabilité qu'un pinceau de section infinitésimale (un rayon) soit occulté par un obstacle unique est égal au ratio de l'aire projetée de l'obstacle et de l'aire projetée du cube d'obstacles, soit :

$${}^I P_{occ}^{ray} = \frac{\pi r_0^2}{L^2}$$

La probabilité de ne pas être occulté est la probabilité contraire car c'est un événement binaire ${}^I P_{ext}^{ray} = 1 - {}^I P_{occ}^{ray}$. Comme nous avons N obstacles placés aléatoirement, la probabilité de n'être occulté par aucun objet est la combinaison des probabilités pour chaque obstacle.

$$\begin{aligned} {}^N P_{vis}^{ray} &= \left[1 - \frac{\pi r_0^2}{L^2} \right]^N \\ {}^N P_{inv}^{ray} &= 1 - {}^N P_{vis}^{ray} \end{aligned}$$

4.3 Pinceau à section constante.

Pour essayer de comprendre ce complexe jeu de paramètres, nous avons dû en éliminer une grande partie. Nous n'avons gardé que la taille des astres ainsi que le nombre et la taille des obstacles. Le but est de comprendre la relation entre classification de visibilité et ces trois paramètres. La section du pinceau est donc homogène sur toute sa longueur, les objets ont tous la même taille et la position des obstacles est uniformément distribuée entre les astres.

Nous utiliserons un ensemble de pinceaux de section carrée constante de différentes tailles afin d'y étudier les classifications de visibilité. Par ailleurs, tous les pinceaux seront alignés avec un axe, si

bien que la longueur d'intersection du pinceau et du cube sera la longueur d'une arête du cube global \mathcal{L} .

Analyse théorique.

Pour simplifier les calculs de probabilités, nous considérerons le pinceau comme cylindrique et les obstacles sphériques. La probabilité qu'un pinceau cylindrique soit complètement occulté par un seul obstacle correspond à la probabilité qu'un rayon intersecte un obstacle réduit de la taille du pinceau. Inversement, la probabilité que le pinceau soit complètement visible est équivalente à la probabilité qu'un rayon ne soit occulté par aucun objet dilaté de la taille du pinceau. (eq. 5.2).

$$\begin{aligned} p_{occ}^{cyl} &= \begin{cases} \frac{\pi(R_p - r_0)^2}{L^2} & , \quad r_0 < R_p \\ 0 & , \quad r_0 > R_p \end{cases} \\ p_{ext}^{cyl} &= 1 - \frac{\pi(R_p + r_0)^2}{L^2} \end{aligned} \quad (5.2)$$

En utilisant la combinaison des probabilités pour un ensemble d'obstacles (eq. 5.1), on obtient :

$$\begin{aligned} p_{inv}^{cyl} &= \left[1 - \frac{\pi(R_p - r_0)^2}{L^2} \right]^N \quad \text{avec } r_0 < R_p \\ p_{vis}^{cyl} &= 1 - \left[1 - \frac{\pi(R_p + r_0)^2}{L^2} \right]^N \\ p_{par}^{cyl} &= 1 - p_{inv} - p_{vis} \end{aligned} \quad (5.3)$$

Vérification pratique.

Nous avons effectué une batterie de tests suivant le protocole décrit précédemment. Les données de densité et taille des obstacles sont utilisées comme paramètres et l'aire de la section du pinceau comme variable. Malgré la réduction des paramètres entrant en jeu, le nombre de données reste relativement élevé pour une visualisation instantanée. Nous avons choisi de présenter une série de courbes qui permettent une lecture précise au détriment d'une lecture plus aisée.

Chaque courbe correspond à un volume d'obstacle et à une densité du cube global. De gauche à droite le volume des obstacles varie de $1 u^3$ à $10^3 u^3$. De haut en bas la densité du cube global varie de 10^{-4} à 10^{-1} . Les tables (tab. 5.6, 5.7) récapitulent les données importantes de cet ensemble de courbes.

Chacune des courbes trace le ratio de réponses pour chaque classification en fonction de l'aire de la section du pinceau qui va de 10^{-2} à $10^4 u^2$ (ou de $5,64 \cdot 10^{-2}$ à $56,42 \cdot 10^2 u$ pour le rayon équivalent).

	volume (u^3)			
	1	10	100	1000
Aire équivalente (u^2)	2,418	11,223	52,094	241,799
Rayon équivalent (u)	0,620	1,336	2,879	6,202

TAB. 5.6: Tailles équivalentes d'obstacles sphériques.

On remarque que les courbes théoriques sont très proches des courbes calculées. Il reste cependant un très petit décalage quasiment constant entre les deux. Cela provient de l'approximation faite dans le calcul théorique où nous considérons les obstacles sphériques alors qu'ils sont cubique dans la scène réelle.

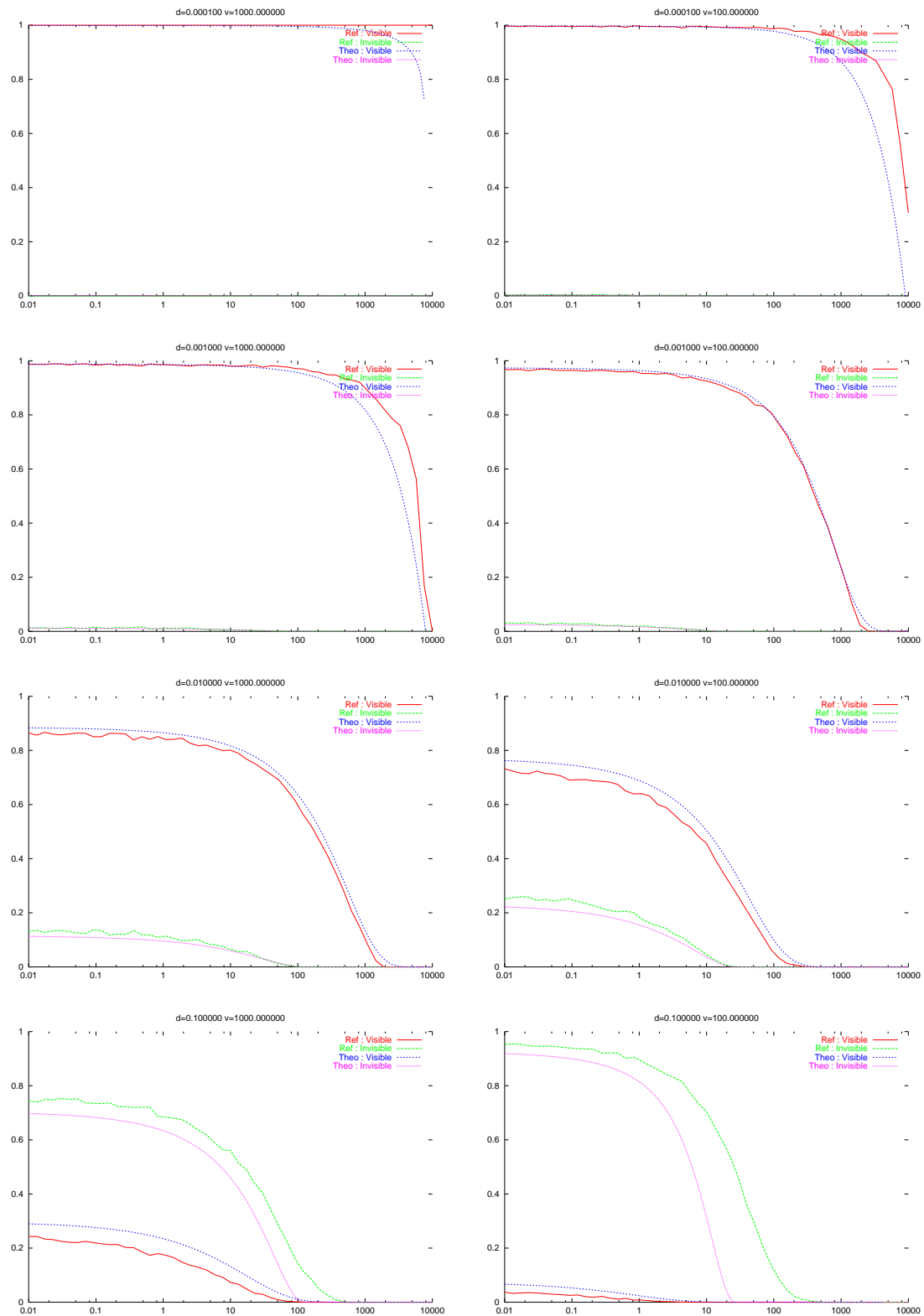


FIG. 5.11: Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau cylindrique. (page 1/2)

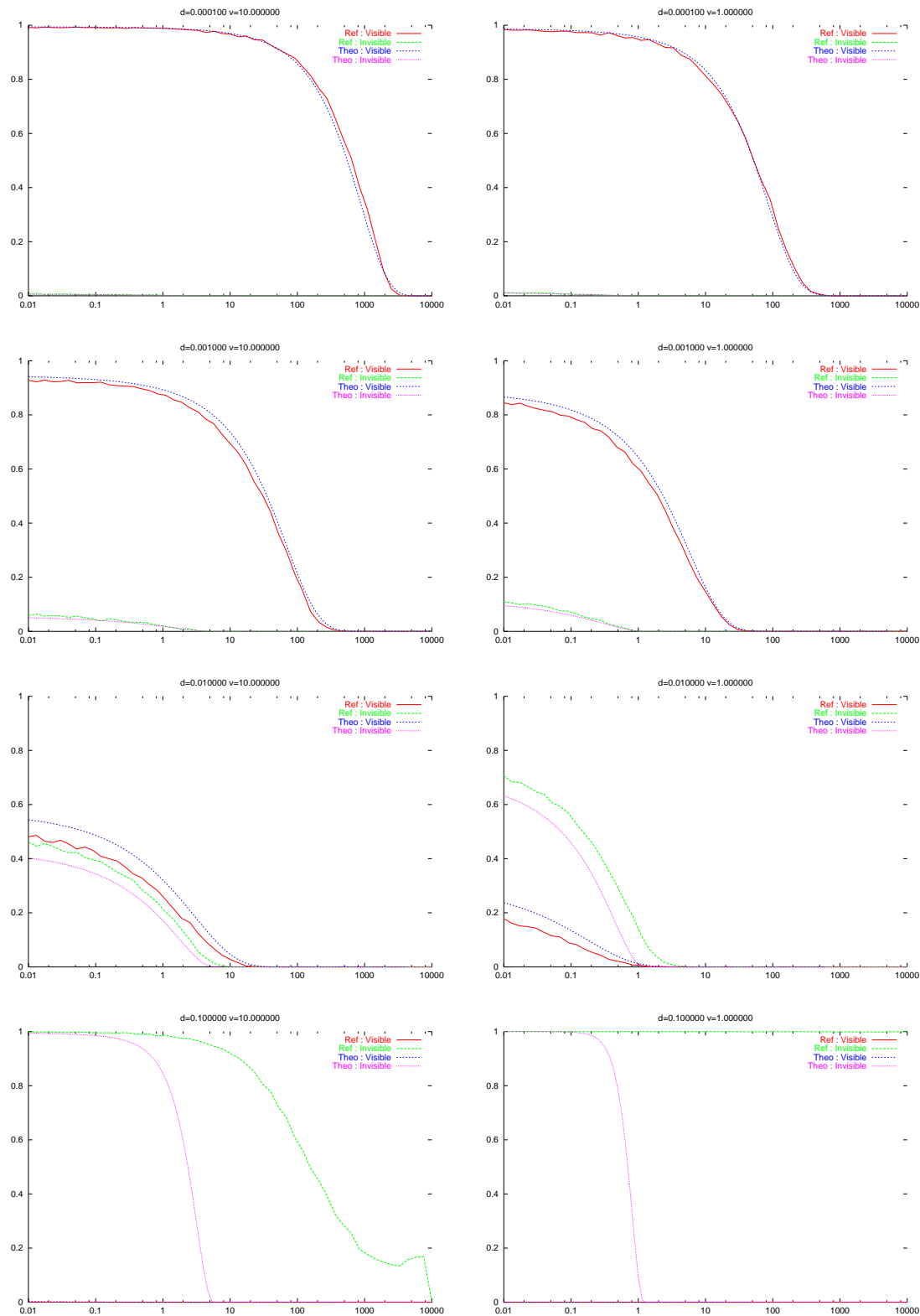


FIG. 5.12: Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau cylindrique. (page 2/2)

		volume (u^3)			
		1	10	100	1000
densité	10^{-4}	100	10	1	0
	10^{-3}	10^3	100	10	1
	10^{-2}	10^4	10^3	100	10
	10^{-1}	10^5	10^4	10^3	100

TAB. 5.7: Table récapitulative du nombre d'obstacles de chaque scène en fonction des paramètres de volume d'obstacles et de densité du cube global.

Un certain bruit apparaît dans les courbes de référence dans les derniers retranchements de nos statistiques, lorsque la densité est forte, les obstacles petits et la section du pinceau importante. Malgré un échantillonnage brutal, quelques petites erreurs peuvent en effet survenir dans ces conditions extrêmes.

Les courbes d'invisibilité s'éloignent fortement pour des densités élevées. Ce phénomène est normal, car les courbes théoriques ne sont calculées que pour l'occultation par un seul obstacle, ce qui devient faux pour des densités élevées et une quantité d'objets importante, où le chevauchement permet de créer une occultation totale générée par plusieurs obstacles. On remarque effectivement que les courbes invisibles théoriques tombent à zéro pour la valeur de l'aire équivalente d'un obstacle.

Ces courbes sont très intéressantes car elles nous permettent de donner une approximation de la classification et donc de pouvoir choisir effectivement un algorithme approprié de discrimination de visibilité.

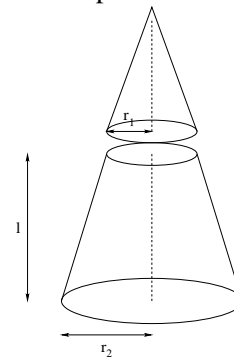
Cependant le nombre de paramètres que nous avons éliminés reste important, il faut continuer les investigations afin de généraliser le plus possible ces résultats. Un des paramètres les plus contraignant est que le pinceau est de section d'aire constante. Notre prochaine étape est de caractériser le comportement de la visibilité à la variation l'aire de la section du pinceau.

4.4 Pinceau à section variable.

Afin de faire varier la section du pinceau, nous avons fixé la taille d'un des éléments et fait varier la taille du second. Les deux centres des astres sont alignés avec un axe, le pinceau théorique résultant est conique. Il coupe le volume d'obstacles en deux sections de pinceau de taille s_1 et s_2 .

Le volume d'un tronc de cône de longueur l , de section circulaire, et dont les rayons extrémités sont r_1 , r_2 est :

$$V_{tc}(r_1, r_2, l) = \frac{l\pi}{3}(r_2^2 + r_1 r_2 + r_1^2)$$



Analyse théorique.

Dans l'ensemble de notre étude nous considérerons les objets comme des disques perpendiculaires à l'axe du pinceau. Ceci évite de prendre en compte le delta de distance (δ) nécessaire à une sphère pour être tangente à un cône (fig. 5.13), terme qui est négligeable pour la plupart des pinceaux.

La probabilité qu'un pinceau conique soit complètement visible dépend du volume du tronc de cône dilaté par la taille d'un objet. Aucun objet ne doit se trouver dans cette zone (eq. 5.4).

$${}^i P_{ext}^{cône}(r_1, r_2) = 1 - \frac{\pi}{3L^2} [(r_2 + r_0)^2 + (r_2 + r_0)(r_1 + r_0) + (r_1 + r_0)^2] \quad (5.4)$$

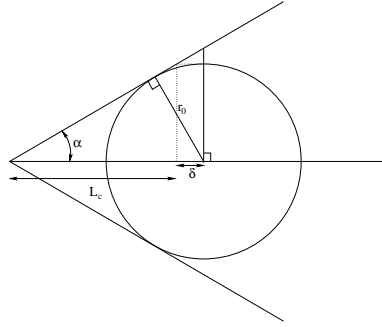


FIG. 5.13: Géométrie de la tangence utilisée entre un disque représentatif d'une sphère et d'un cône (représentation en coupe).

La probabilité d'occultation n'est pas aussi directe. La zone d'occultation totale par un obstacle de taille r_0 est un cône inversé par rapport au cône du pinceau. Nous utilisons le calcul d'une distance critique L_c , distance entre les sommets des deux cônes et L_1 la distance du tronc de cône à son sommet (fig. 5.13).

$$L_c(r_1, r_2) = \frac{Lr_0}{|r_2 - r_1|} \quad , \quad L_1(s_1, s_2) = L \frac{r_1}{r_2 - r_1}$$

Ainsi le volume d'occultation totale est l'intersection entre le cône inversé et le volume des obstacles. Trois cas sont possibles selon que le sommet se trouve à l'extérieur de la zone d'obstacles du petit coté du pinceau ou du grand coté du pinceau, ou dans la zone d'obstacles (fig. 5.14).

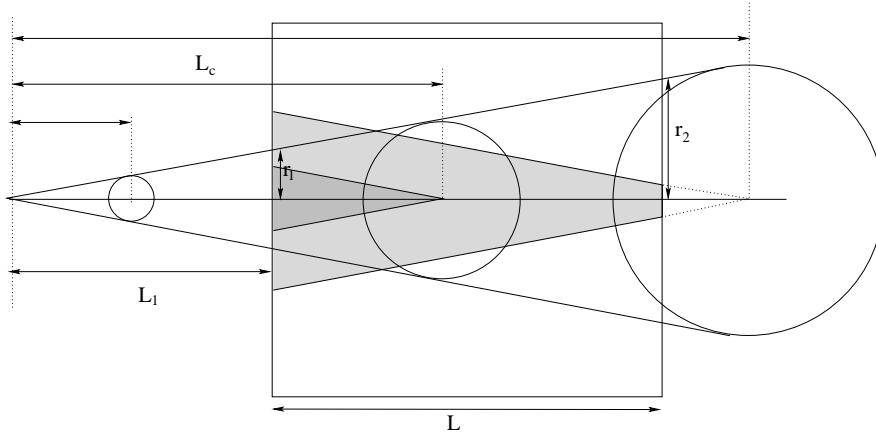


FIG. 5.14: Volume d'occultation totale dans un pinceau conique.

$${}^i P_{occ}^{cône} = \begin{cases} 0 & , \quad L_c \in [0, L_1] \\ \frac{\pi}{3L^2(r_2 - r_1)} (r_0 - r_1)^3 & , \quad L_c \in [L_1, L + L_1] \\ \frac{\pi}{3L^2} [(r_0 - r_1)^2 + (r_0 - r_1)(r_0 - r_2) + (r_0 - r_2)^2] & , \quad L_c \in [L + L_1, \infty] \end{cases} \quad (5.5)$$

Vérification pratique.

Nous avons effectué le même protocole de tests que dans le cas des pinceaux cylindriques en faisant varier la taille d'un des astres et en laissant la seconde fixe. La taille de la section du pinceau et de $5 u^2$ à une extrémité et varie de 10^{-2} à $10^4 u^2$ pour la seconde.

Les conclusions sont identiques à celles obtenues pour un pinceau cylindrique. La distance entre les courbes théoriques et pratiques est très faible pour une majorité des couples densité / taille des obstacles. À forte densité, quand apparaît de l'occultation combinée, la théorie n'arrive pas à prédire ce qui engendre une différence croissante en fonction de ces deux paramètres tout comme dans le cas précédent (fig. 5.15).

4.5 Application à un cas concret.

Les formulations théoriques que nous avons décrites à partir des statistiques de visibilité permettent de prédire la probabilité de classification de la visibilité entre deux axes. La prédiction étant valable même pour un petit nombre d'obstacles, nous allons vérifier son application sur des scènes réelles dans le cadre de la radiosité hiérarchique.

La valeur probable de la classification de visibilité nous permet d'utiliser un outil de classification plus performant et/ou précis pour une classification particulière. Les volumes de vision étendus sont le seul outil à notre disposition pour déterminer la classification INVISIBLE mais son évaluation est relativement coûteuse. La prédiction du cas INVISIBLE nous permet d'utiliser les volumes de vision dans les cas les plus probables de son apparition.

Calcul des paramètres des formules statistiques.

Les formulations dépendent de la longueur de traversée du volume d'obstacles (L), de la taille de la section du pinceau aux frontières du volume (r_1, R_2) ainsi que de la taille et du nombre des obstacles.

Dans les formulations des statistiques que nous avons développées, la taille des obstacles est constante. Ce qui n'est pas le cas dans une cellule quelconque de la hiérarchie d'une scène. Dans le cas général, la taille caractéristique des obstacles dépend aussi de l'angle avec lequel le pinceau traverse le volume. En approximation nous utilisons la moyenne des surfaces des polygones de la cellule pondérées par la probabilité moyenne du coefficient de projection sur une hémisphère [SP94] :

$$\frac{1}{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} \int_{\phi=-\pi}^{\pi} \cos \theta \sin \phi \, d\phi d\theta = \frac{1}{2}$$

On obtient donc la valeur moyenne de l'aire projetée des polygones :

$$s_0 = \frac{1}{2N} \sum_{i=1}^N \mathcal{A}_i$$

où \mathcal{A}_i la surface du polygone i et N le nombre de polygone dans la cellule.

Outil de prédiction.

Nous avons créé, dans le schéma d'un graphe de raffinement (cf. chapitre 4, page 69), un outil de prédiction probable de la classification de visibilité. En fonction de la configuration géométrique, l'outil calcule les paramètres impliqués dans les formules d'estimation probabilistes et retourne la classification la plus probable. Ces valeurs de retour sont les classifications PROBABLEMENT PARTIEL, PROBABLEMENT VISIBLE et PROBABLEMENT INVISIBLE.

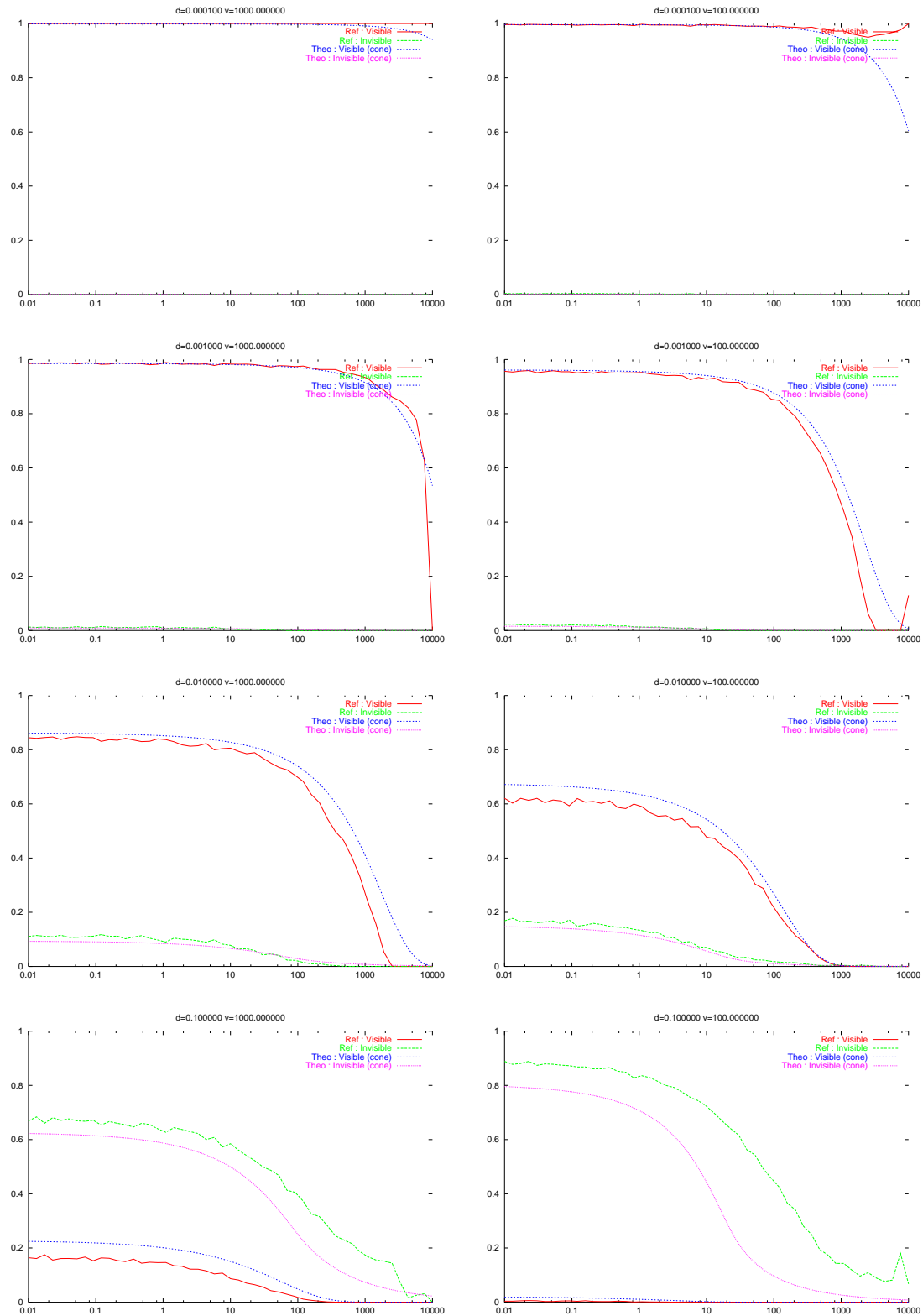


FIG. 5.15: Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau conique. (page 1/2)

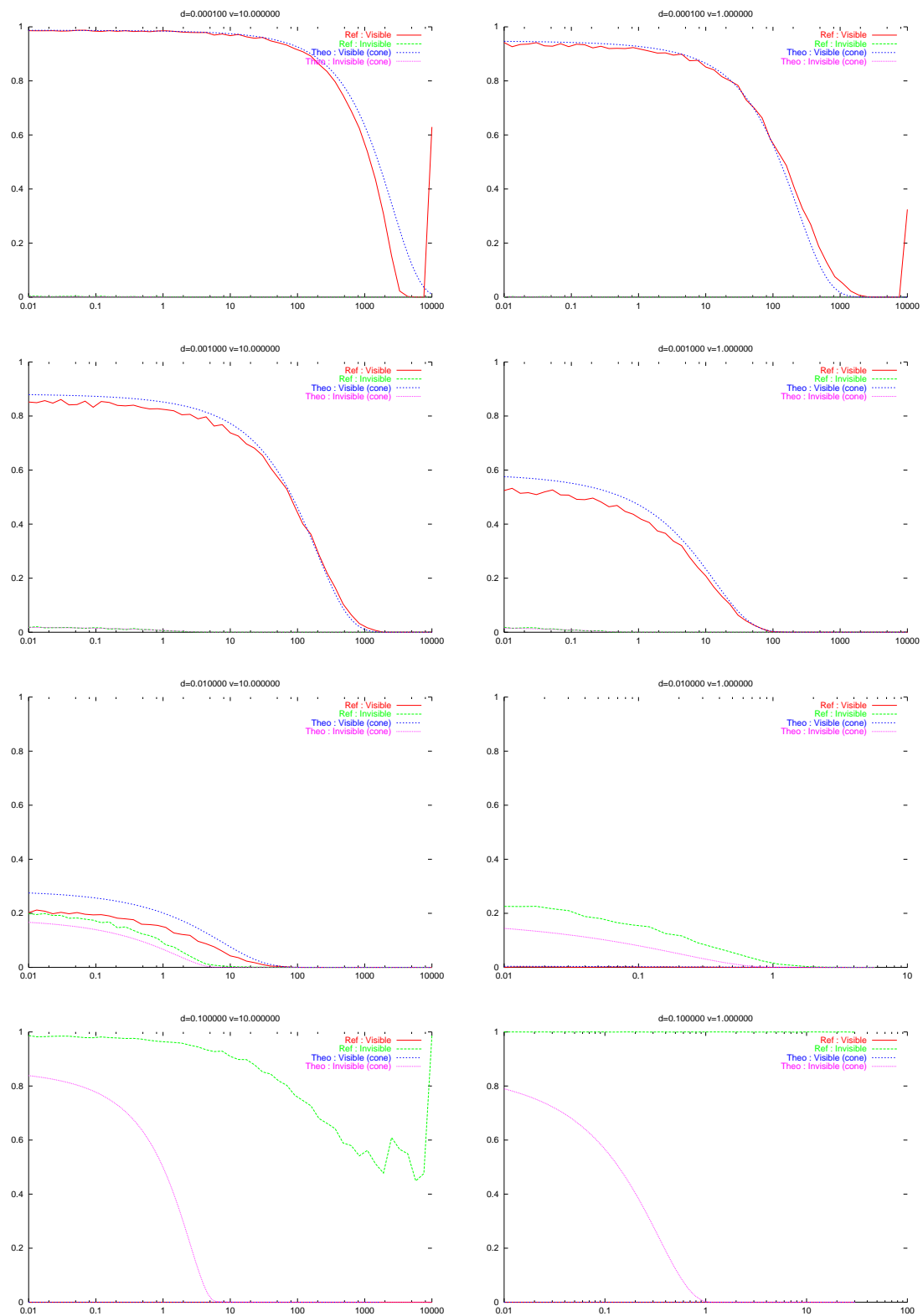


FIG. 5.16: Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau conique. (page 2/2)

L'outil doit déterminer l'ensemble des cellules-obstacles qui coupent la section du pinceau intégralement. Ceci est obtenu par un algorithme de volume de vision. La classification *VISIBLE* est donc une réponse possible de l'outil.

L'ensemble des obstacles potentiels (et pas seulement les obstacles qui coupent la section du pinceau) est réutilisé par les outils suivants de classification et d'évaluation de visibilité grâce à l'utilisation du gestionnaire de données (cf. §3.1, page 75).

Résultats.

Nous avons utilisé deux oracles de raffinement simple basé sur le critère BF (fig. 4.4). Le premier utilise une classification par volume de vision étendus systématique, le second utilise l'outil de prédiction afin de diriger l'utilisation de l'outil de classification par volumes de vision étendus.

L'évaluation des résultats n'est pas évidente car les outils de classification réutilisent la liste des obstacles potentiels créée par l'outil de prédiction. De plus le raffinement est légèrement modifié car certains cas d'invisibilité totale systématiquement déterminés dans le premier raffineur ne le sont plus par le second.

Nous avons choisi trois critères de jugement :

- La proportion effective de réponses *INVISIBLE* déterminée par l'outil utilisant les volumes de vision étendus ;
- Le temps de raffinement ;
- Un critère subjectif de qualité visuelle de la solution calculée.

La scène de tests utilisée est la scène du labyrinthe comportant une quantité importante de grands obstacles (fig. 5.7b). La scène de tests du labyrinthe possède relativement peu de polygones, ce qui la place en limite des statistiques car peu de processus de raffinement sont exécutés : de l'ordre de 250.000 pour la première itération et 100.000 pour la seconde. Elle possède par contre de grands obstacles permettant d'avoir un nombre de classifications *INVISIBLE* représentatif.

Les résultats visuels sont très comparables avec une très légère dégradation en utilisant l'outil de prédiction. Il est vrai que de nombreuses approximations sont faites quant à la taille des obstacles.

Grâce à l'utilisation d'un outil utilisant les volumes de vision étendus guidé par un outil de prédiction de la visibilité, nous passons de 15% de classifications *INVISIBLE* à 63%. L'outil de volumes de vision étendus est utilisé plus spécifiquement dans les cas probables d'invisibilité totale et permet de gagner sur son temps important d'évaluation de la visibilité.

La classification *INVISIBLE* est la classification la plus difficile à obtenir et les algorithmes qui arrivent à la déterminer sont très coûteux. Une caractérisation rapide des configurations où il est probable ou non que les astres soient complètement occultés permet de gagner temps. Dans la scène du labyrinthe utilisée, le nombre de classifications *INVISIBLE* est assez élevé de par les grands obstacles transversaux. L'utilisation de l'outil de prédiction permet, dans ce cas favorable, de gagner près de la moitié du temps de calcul pour chaque itération (tab. 5.8).

En testant les résultats théoriques, limité à ces paramètres de densité et de forme de pinceau, sur une scène concrète permet de montrer que l'extrapolation de cette théorie fournit déjà un gain de temps avec très peu de perte de qualité. L'outil de prédiction permet d'aiguiller le raffinement vers l'outil de classification de visibilité le mieux adapté.

4.6 Conclusion.

La visibilité entre deux astres quelconques dans un volume d'obstacles, eux aussi quelconques, est un problème extrêmement complexe à résoudre pour un ordinateur lorsque l'on utilise une description surfacique de l'environnement.

Utilisation de l'outil utilisant les volumes de vision étendus de façon systématique :

	:1 ^{ère} itération	2 ^{nde} itération
Temps (s.) :	165	54
Liens :	76.900	86.400
VISIBLE :	57%	79%
INVISIBLE :	15%	6%
PARTIEL :	28%	15%

Utilisation de l'outil de prédiction en tête de graphe de raffinement :

	:1 ^{ère} itération	2 ^{nde} itération
Temps (s.) :	73	32
Liens :	80.200	90.400
VISIBLE :	0%	10%
INVISIBLE :	63%	12%
PARTIEL :	37%	78%

TAB. 5.8: Résultats de l'outil de prédiction de la classification de visibilité basée sur une étude statistique sur la distribution des obstacles.

Nous avons choisi de prendre l'optique d'une étude statistique afin d'essayer d'en définir le comportement en fonction de certains paramètres. Notre étude particulière basée sur la densité des obstacles et la forme du pinceau intersectant leur volume permet de définir un comportement théorique très proche du comportement statistique pratique.

5 Bilan.

La visibilité reste un des goulots d'étranglement des algorithmes qui doivent l'utiliser. Nous avons proposé dans ce chapitre, une étude de la visibilité appliquée à l'algorithme de radiosité hiérarchique qui ne peut faire aucune hypothèse simplificatrice quant aux configurations géométriques. Dans un premier temps, nous avons présenté la classification de visibilité utilisée ainsi que les différentes réponses possibles des outils de classification.

Nous nous sommes penché sur les deux principaux outils de classification utilisés par la radiosité : l'outil d'échantillonnage et l'outil de volumes de vision. Nous y étendons l'outil classique de volume de vision afin de pouvoir certifier la classification INVISIBLE, et dans le même esprit que l'algorithme initial, nous y présentons comment créer ces *volumes de vision étendus* d'une façon optimisée.

La classification de visibilité par échantillonnage est la plus utilisée car elle est facile à réaliser et permet d'obtenir une estimation du facteur de visibilité. Au cours du calcul de la solution, de nombreux rayons sont lancés afin de tester leurs intersections avec les obstacles de la scène. Nous proposons un algorithme d'optimisation de l'intersection de rayons en parcourant l'arbre de visibilité à partir du dernier obstacle intersecté. Ceci afin de tirer parti de la cohérence géométrique des obstacles au sein de la hiérarchie.

Les gains de temps de cette optimisation dépendent beaucoup de la scène utilisée et sa mise en œuvre n'est pas aisée. Nous préconisons plutôt l'usage d'un cache de rayons à plusieurs niveaux.

Le raffinement est un processus récursif dont les volumes de vision des fils sont inclus dans ceux de leur ascendance. Nous utilisons cette propriété afin de sauvegarder un état des obstacles potentiels

au sein de l'arbre de visibilité, et ainsi réduire le nombre de tests d'intersections pour les descendants. La liste d'optimisation proposée garanti une accélération des tests de visibilité pour une occupation mémoire fixe définie par l'utilisateur.

Dans un dernier point, nous avons effectué une étude statistique de la visibilité entre deux éléments au travers d'un volume d'obstacles de propriétés de densité et de répartition d'obstacles connues. De cette étude nous avons défini un comportement théorique qui nous permet d'obtenir à très peu de frais une approximation de la classification de la visibilité. Cette évaluation nous permet de choisir un outil de classification de visibilité adapté à la configuration locale des obstacles.

Reconstruction

Il existe pour chaque problème complexe une solution simple, directe et fausse.

H.L. Mencken

L'algorithmme de radiosité hiérarchique caractérise la solution de radiosité par l'intermédiaire des liens. Ils sont établis à différents niveaux hiérarchiques et représentent le transport énergétique entre un émetteur et un récepteur. La solution est une fonction définie par morceaux au niveau de la subdivision la plus fine de la géométrie.

La solution réelle d'éclairage de la scène est essentiellement continue, seules les singularités aux jointures de la géométrie créent des discontinuités ponctuelles. Il est raisonnable de recréer artificiellement la continuité de la solution à partir de la représentation discrète que l'on connaît d'elle.

Nous allons voir les méthodes de reconstruction couramment utilisées en radiosité hiérarchique ainsi que les problèmes posés par chacune. Ensuite nous proposerons deux solutions pour faire un compromis entre temps de calcul et qualité du résultat. Nous finirons par la comparaison des résultats entre les différentes méthodes.

1 Reconstruction de la solution.

Le raffinement établit les liens au niveau hiérarchique qu'il juge opportun en fonction de la base de fonctions utilisée dans l'algorithmme. La phase de collecte en radiosité hiérarchique permet de calculer l'irradiance sur chacun des éléments. Ensuite l'irradiance est convertie en radiosité. Le principe

de séparer la collecte et le push/pull rend la contribution énergétique indépendante de la direction d'émission.

Nous verrons que, dans le cas du regroupement de surfaces, cette simplification n'est pas sans inconvénients. Elle peut être outrepassée par une modification de la collecte appelée *collecte immédiate* (*immediate push* et *illumination sample* [SSG⁺99]).

À la reconstruction, s'ajoute souvent une phase d'interpolation de la solution qui élimine ses discontinuités. *L'interpolation finale*, la plus fréquemment utilisée, n'est en fait qu'une modification de la visualisation et ne rentre pas directement en compte dans la reconstruction de la solution.

Nous finirons par décrire un type de reconstruction qui permet de générer des images fixes d'une bonne qualité mais, dans ce cas, la qualité de la globalité de la solution n'est pas modifiée.

1.1 Collecte immédiate.

Dans l'algorithme de radiosité hiérarchique sans regroupement de surfaces, une fois la collecte effectuée, chaque élément connaît l'irradiance qu'il reçoit et oublie la notion de localisation de l'émission. Dans le cas précis de la radiosité hiérarchique avec regroupement de surfaces, l'irradiance des regroupements ne peut pas être considérée comme indépendante de la direction d'émission.

Contrairement à une subdivision de surfaces, la répartition des orientations des fils d'un groupe de surfaces est probablement très hétérogène. Il serait trop imprécis de considérer l'apport énergétique identique pour toutes les surfaces le composant. Il faut prendre en compte l'angle entre la direction d'émission et chacun des récepteurs.

Pour tenir compte de l'orientation des surfaces dans un regroupement, il faut connaître la localisation de l'émission. Cette localisation peut être effectuée soit en sauvegardant les liens et en recalculant une valeur d'irradiance pendant le push/pull (valeur qui est censée avoir été calculée durant la collecte), soit en propageant la contribution d'un lien (émetteur) aux éléments hiérarchiques inférieurs lors de la collecte. Connaissant l'émetteur, il est alors possible de prendre en compte l'orientation des surfaces.

Si la contribution des liens est prise en compte immédiatement, le coût induit est fonction du nombre d'éléments moyens impliqués dans chaque échange énergétique. Plus les liens seront placés haut dans la hiérarchie, plus le surcoût de calcul sera important. Dans le cas de l'*illumination sample*, l'ensemble des liens est sauvegardé durant la phase de collecte, ce qui permet de ne pas augmenter la complexité mais augmente l'espace mémoire.

1.2 Interpolation finale.

Une fois le raffinement terminé, la solution est représentée par un ensemble de liens établis à des niveaux hiérarchiques différents. Pour conserver la cohérence de la solution au travers de la hiérarchie, à chaque itération, il faut propager l'information de lumière au travers de la scène et de la solution de radiosité par le biais des liens. Puis on peut lisser la fonction de radiosité par une passe d'interpolation.

La technique de reconstruction classique est une simple interpolation des valeurs de radiosité aux sommets des éléments hiérarchiques les plus fins (surfaces subdivisées). Elle a l'avantage de tirer parti des performances actuelles des cartes graphiques, capables d'afficher des polygones dont la couleur est interpolée à partir des sommets. Ceci permet une visualisation interactive pour un coût supplémentaire de calculs très faible. Le processus total de reconstruction s'effectue en quatre étapes :

- collecte : traversée descendante de la hiérarchie avec calcul de l'irradiance au niveau de chaque élément hiérarchique (fig. 6.1b) ;
- push : traversée descendante de la hiérarchie et transformation de l'irradiance en radiosité (fig. 6.1c) ;

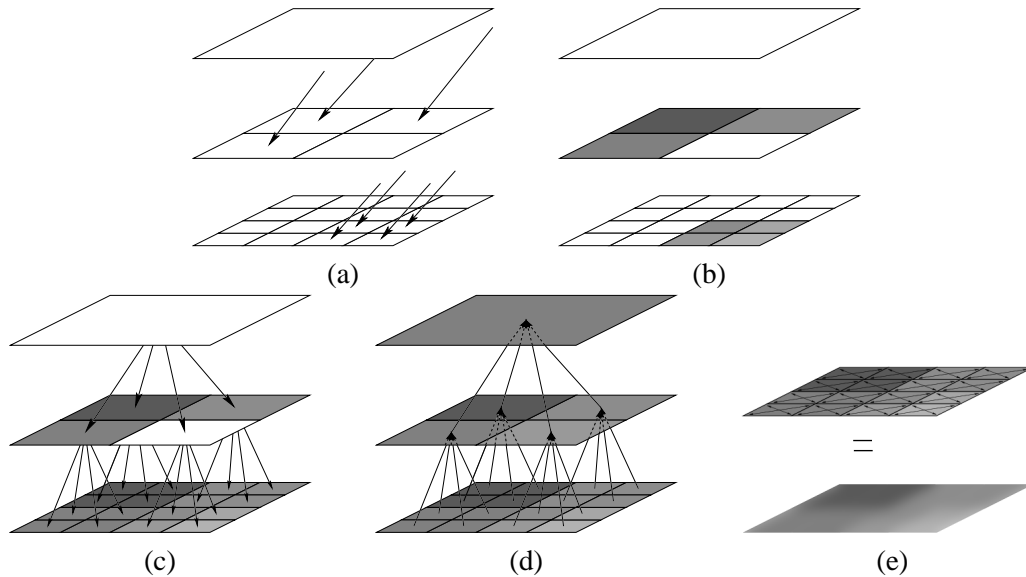


FIG. 6.1: Conservation de la cohérence de la hiérarchie et reconstruction par interpolation finale. (a) niveau d'établissement des liens. (b) collecte. (c) push. (d) pull. (e) interpolation finale.

- pull : traversée ascendante de la hiérarchie avec mise à jour de la radiosité de tous les éléments de la hiérarchie (fig. 6.1d). Cette étape n'agit pas directement sur la reconstruction de la fonction mais est nécessaire pour le transfert d'énergie durant une itération ;
- interpolation finale (*smooth shading*) : parcours des feuilles de la hiérarchie et transfert de la radiosité des surfaces aux sommets (fig. 6.1e).

Il n'y a pas de rapport direct entre l'interpolation et les liens. De par ce fait, la fonction de radiosité interpolée est dépendante de la résolution géométrique de la solution.

Dans l'exemple de la figure 6.2, nous avons deux solutions identiques au niveau de l'établissement des liens, mais dont les résolutions géométriques sont différentes. Une valeur de radiosité est calculée à chaque sommet comme moyenne des valeurs de radiosité des éléments hiérarchiques terminaux contigus. Pour un même ensemble de liens, l'interpolation finale donne des résultats différents en fonction de la subdivision géométrique. Cette interpolation n'est pas toujours très proche de ce que l'on peut attendre (fig. 6.2d).

Dans tout ce chapitre, nous allons utiliser un schéma de représentation des algorithmes de collecte et de push/pull basé sur le parcours de la hiérarchie (fig. 6.3). Un parcours complet de la hiérarchie est composé d'une phase d'initialisation des éléments de la hiérarchie (a), d'une descente de père en fils jusqu'aux éléments terminaux et de la remontée de la hiérarchie. À chaque élément traité durant la descente, une action peut être exécutée (b), une particulière peut être exécutée pour chaque élément terminal (c). La remontée de la hiérarchie est basée sur le même principe (d).

Théoriquement, les phases de push/pull et la phase d'interpolation finale sont complètement indépendantes. En pratique, pour éviter de faire trop de parcours complets de la hiérarchie, l'interpolation finale est réalisée au cours du traitement des éléments terminaux pendant le push/pull (fig. 6.4).

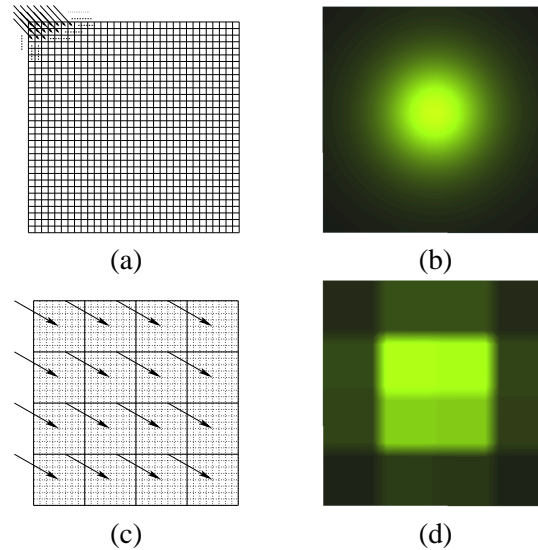


FIG. 6.2: Variation de l'interpolation finale en fonction de la hauteur d'établissement des liens dans la hiérarchie. (a) établissement des liens au niveau de la géométrie. (b) solution reconstruite par interpolation finale de (a). (c) établissement des liens à des niveaux hiérarchiques supérieurs à la géométrie. (d) solution reconstruite par interpolation finale de (c).

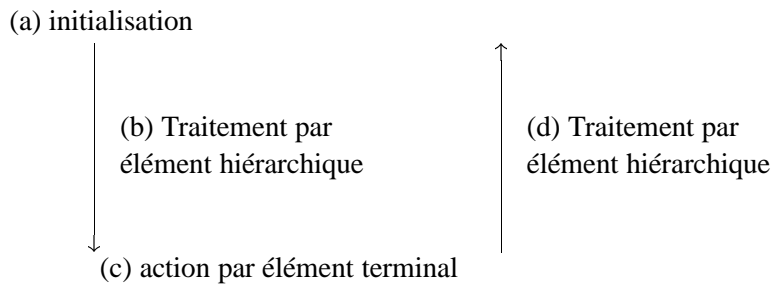


FIG. 6.3: Schématisation d'un parcours hiérarchique et des différents traitements applicables. Avec la convention d'utiliser \leftarrow donnée lue et \rightarrow donnée affectée ($()$) pour symboliser respectivement l'accès en lecture et en écriture à une donnée.

1.3 Collecte par pixel.

La *collecte par pixel* (*final gathering*) permet de créer une image fixe de qualité supérieure à ce que peut fournir l'algorithme de radiosit  seul. Le principe est de calculer une valeur de radiosit  pr cise en tous les points visibles de l'image.

La collecte par pixel fut utilis e dans le premier algorithme d' clairage global en deux passes : une passe de radiosit  et une passe de lancer de rayons (cf. §4.7, page 22). La passe de lancer de rayons calcule l' clairage direct et les effets sp culaires purs.

L'image obtenue est de qualit  sup rieure – au sens du rendu photor aliste – mais ceci   un c t de calcul tr s important.   chaque pixel correspond une surface visible o  un ensemble de liens

Collecte :

Mise à zéro de l'irradiance

→ *Irradiance (mise à zéro)*

Par élément hiérarchique :

Calcul de l'irradiance

→ *Irradiance (accumulation)*← *Radiosité de surface*

0



0

push/pull :

Mise à zéro de la radiosité des sommets

→ *Radiosité sommet (mise à zéro)*

Par élément hiérarchique :

Somme des irradiances

Moyenne des radiosités :

→ *Radiosité surface (affectation)*

Conversion de la somme d'irradiance en radiosité :

← *Irradiance*← *Émittance*→ *Radiosité surface (affectation)*→ *Radiosité point (accumulation)***FIG. 6.4:** Description de la collecte et du push/pull de l'interpolation finale.

influe sur son éclairage. À chacun de ces liens, il faut recalculer les échanges lumineux (facteur de forme + visibilité). De plus, une seule image est créée ; la visualisation interactive n'est pas améliorée comme dans le cas de l'interpolation finale.

2 Améliorations de la reconstruction.

Nous cherchons à améliorer la qualité de la visualisation interactive de la solution. Les temps de calculs de la collecte par pixel sont trop importants et l'interpolation finale crée des artefacts dépendant de la géométrie. Nous voulons obtenir une reconstruction de la solution dans des temps raisonnables, visualisable interactivement après calcul et indépendante de la résolution géométrique.

Nous allons montrer que l'approche la plus simple ne suffit pas à obtenir des résultats visuellement corrects, puis nous proposerons deux méthodes de reconstruction basées sur un calcul plus précis de l'irradiance : l'une au niveau des éléments hiérarchiques, l'autre au niveau de leurs sommets.

2.1 Interpolation hiérarchique.

Afin de régler le problème posé par l'interpolation finale (cf. §1.2, page 128) et éliminer la dépendance à la résolution géométrique, il faut interpoler les échanges énergétiques au niveau des liens et non au niveau des éléments terminaux.

Dans la phase de push habituelle, l'irradiance des liens est considérée constante sur l'élément et toutes ses subdivisions. Nous voulons descendre une valeur interpolée d'irradiance sur l'élément en lui même et son voisinage proche, ceci en tenant compte du niveau hiérarchique d'établissement du lien. De plus, nous ne voulons pas perdre ou ajouter de l'énergie, ni créer des discontinuités durant cette phase de reconstruction.

L'interpolation avec le voisinage impose d'utiliser un noyau d'interpolation qui a les propriétés suivantes :

- valeur centrale d'interpolation culminant à 1 de façon à avoir la valeur d'irradiance calculée maximale au centre de l'élément ;
- la somme des noyaux adjacents doit être constante de valeur 1 afin de conserver la quantité d'énergie calculée ;
- le noyau doit être symétrique par rapport au centre afin de ne pas créer de motifs répétitifs.

En 2 dimensions, l'ensemble des noyaux correspondants est l'ensemble des fonctions centralement symétriques par rapport au point $(\frac{1}{2}; \frac{1}{2})$ (eq. 6.2).

Soit f la fonction d'interpolation d'un élément hiérarchique avec son voisinage. Les propriétés précédentes permettent de définir f par :

$$f : \begin{cases} \forall x \in]-\infty; -1] \cup [1; \infty[, f(x) = 0 \\ f(0) = 1 \\ f(x) = -f(-x) \\ f(x) + f(x-1) = 1 \end{cases}$$

En réduisant l'étude de la fonction sur $[0; 1]$, nous allons prouver que la fonction noyau est centralement symétrique par rapport au point $(\frac{1}{2}; \frac{1}{2})$. Avec $x = \frac{1}{2} + x$, la conservation de l'énergie donne :

$$\begin{aligned} f(\frac{1}{2} + x) + f(x - \frac{1}{2}) &= 1 \\ \Leftrightarrow f(\frac{1}{2} + x) &= 1 - f(x - \frac{1}{2}) \\ \Leftrightarrow f(\frac{1}{2} + x) &= 1 - f(\frac{1}{2} - x) \end{aligned} \quad (6.1)$$

Soit g la fonction obtenue par changement de repère en $(\frac{1}{2}; \frac{1}{2})$ de la fonction f :

$$\begin{aligned} g(x) &= f(x + \frac{1}{2}) - \frac{1}{2} \\ g(x) &= f(x + \frac{1}{2}) - \frac{1}{2} \\ &= 1 - f(\frac{1}{2} - x) - \frac{1}{2} \\ &= \frac{1}{2} - f(\frac{1}{2} - x) \\ &= -g(-x) \end{aligned} \quad (6.2)$$

g est une fonction impaire, f est donc une fonction centralement symétrique par rapport au point $(\frac{1}{2}; \frac{1}{2})$. Nous avons pris comme exemple un noyau simple avec $f(x) = 1 - x$. Ce noyau est une fonction triangle en 2 dimensions et pyramidale en 3 dimensions (fig. 6.5).

Sur les bords de la fonction, quand il n'y a pas de voisinage, la fonction d'interpolation doit être constante et de valeur 1 car elle est la seule à avoir une influence. La valeur nulle d'interpolation est atteinte au centre du voisinage proche afin de conserver la valeur centrale du voisin. Dans un contexte hiérarchique, cela implique la subdivision de l'élément voisin pour que son centre ait une existence géométrique. Nous avons choisi de forcer la subdivision dans le cas d'éléments hiérarchiques subdivisibles et de ne pas faire d'interpolation pour les éléments terminaux non subdivisibles.

Hélas, cette interpolation crée des artefacts incontournables quand elle est utilisée avec une hiérarchie. Comme le montre l'exemple de la figure 6.6, la reconstruction d'une fonction constante établie

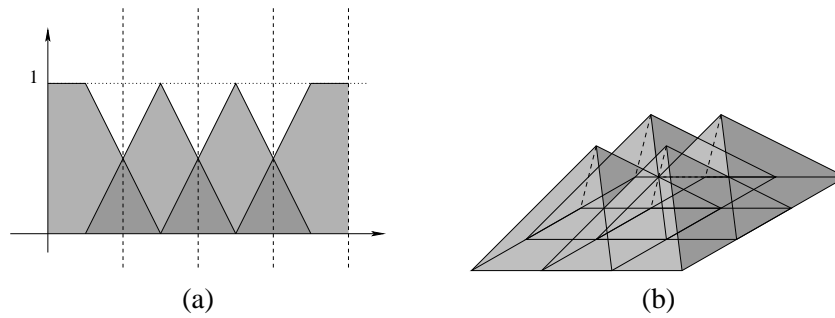


FIG. 6.5: Noyau d'interpolation symétrique permettant de conserver les deux propriétés de valeur centrale et de conservation de l'énergie. (a) 2 dimensions. (b) 3 dimensions.

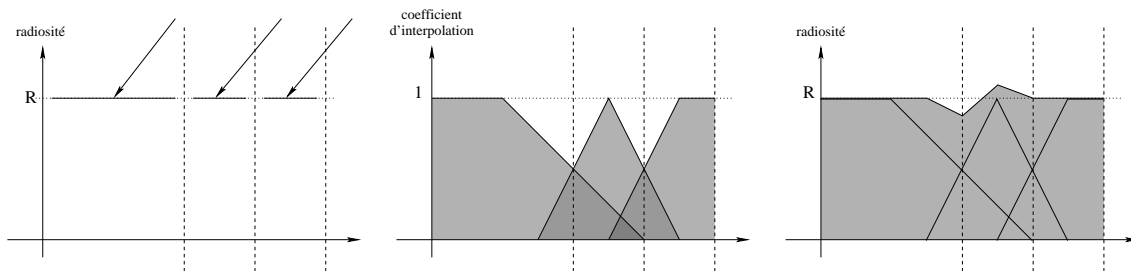


FIG. 6.6: Reconstruction en 2 dimensions par interpolation d'une fonction constante dont les valeurs sont définies à des niveaux hiérarchiques différents.

à des niveaux hiérarchiques différents, n'est plus constante, les propriétés désirées étant pourtant respectées (fig. 6.6).

Ce comportement n'est pas acceptable dans le cadre de la radiosité car les effets visuels induits sont désagréables. Le phénomène est, de plus, amplifié par la sensibilité de l'oeil aux variations de la dérivée spatiale de la luminance.

Les deux prochaines sections décrivent des algorithmes de reconstruction qui créent une interpolation implicite sans artefacts visuels. Nous terminerons par une étude de l'intégration de la visibilité dans ces algorithmes.

2.2 Collecte immédiate généralisée.

La collecte immédiate généralisée est l'extension de la collecte immédiate aux surfaces subdivisées. Au cours de la collecte, lorsque l'on connaît encore la notion d'irradiance directionnelle, on calcule, au niveau des subdivisions, une irradiance plus précise d'une surface que celle fournie par le lien en lui-même sur l'élément non subdivisé.

Cela augmente le nombre de calculs effectués durant la collecte car il faut recalculer un facteur de forme à chaque subdivision impliquée dans l'échange énergétique. Le surcoût est fonction du niveau hiérarchique d'établissement des liens (fig. 6.7).

La valeur d'irradiance étant recalculée pour toutes les subdivisions impliquées, il n'est pas nécessaire de la stocker sur le lien. Par contre, afin d'éviter un recalcul coûteux de la visibilité, celle-ci peut être stockée.

Collecte :

Mise à zéro de l'irradiance

→ *Irradiance (mise à zéro)*

Par lien :

0

0

Calcul de l'irradiance

→ *Irradiance (accumulation)*← *Radiosité de surface*← *Émittance*

0

push/pull : (identique au push/pull normal)

Mise à zéro de la radiosité des sommets

→ *Radiosité sommet (mise à zéro)*Par élément hiérarchique :
Somme des irradiances

Moyenne des radiosités :

→ *Radiosité surface (affectation)*

Conversion de la somme d'irradiance en radiosité :

← *Irradiance*← *Émittance*→ *Radiosité surface (affectation)*→ *Radiosité point (accumulation)***FIG. 6.7:** Description de la collecte et du push/pull de la collecte immédiate généralisée.

Cette méthode est facile à intégrer dans un algorithme de radiosité hiérarchique avec regroupement car il suffit d'appliquer le principe de collecte immédiate effectué aux groupes de surfaces, ceci pour un surcoût de temps de calcul acceptable pour la plupart des applications (cf. §3.2, page 136).

2.3 Collecte par sommet.

Pour aller encore plus loin dans le concept de calcul d'irradiance et d'interpolation, nous avons calculé directement les valeurs aux sommets dans la phase de collecte immédiate. L'interpolation n'est alors plus nécessaire mais il faut en revanche modifier le pull afin de mettre à jour la valeur de radiosité des surfaces (fig. 6.8).

Chaque sommet appartient à plusieurs éléments hiérarchiques. Les liens sont stockés sur des éléments et non pas directement sur les sommets. Ce qui implique qu'à chaque collecte par sommet, les valeurs d'irradiance sont recalculées plusieurs fois au même point. Il est difficile d'éviter cette répétition car il faudrait alors sauvegarder pour tous les sommets l'ensemble des liens qui ont déjà

Collecte :

Mise à zéro de la radiosité des sommet

→ *Radiosité sommet (mise à zéro)*

Par lien :

0

0

Calcul de radiosité :

0

→ *Radiosité sommet (accumulation)*← *Radiosité surface***Push/Pull :**

0

0

Moyenne des radiosités :

→ *Radiosité surface (affectation)*

Ajout de l'émittance aux sommets :

← *Émittance*→ *Radiosité point (accumulation)***FIG. 6.8:** Description de la collecte et du push/pull de la collecte par sommet.

calculé la valeur d'irradiance. Cela limite fortement son utilisation car les temps de calcul sont très élevés pour des résultats visuels équivalents à la collecte immédiate généralisée (cf. §3, page 136).

2.4 Liens et visibilité.

Chaque type de reconstruction proposé peut être associé à un type de lien différent. Ceci permet de mélanger les différentes approches de reconstruction au sein même de la reconstruction de la fonction de radiosité. Seules les étapes de collecte et de push/pull sont modifiées.

Dans les deux types de reconstruction proposés, il est possible d'intégrer ou non la visibilité. Les liens qui impliquent le recalcul de la visibilité ne nécessitent aucun stockage contrairement aux autres types de liens qui doivent stocker la visibilité – tout du moins dans les cas de classification PARTIEL de la visibilité (cf. §1.1, page 98) –.

Si l'on ne stocke pas la visibilité lors du calcul de la solution, il est possible de choisir la méthode de reconstruction après coup. Pour cela, il faut repropager la lumière dans la scène initiale sans refaire de processus de raffinement. Ceci est une méthode d'amélioration de la solution à posteriori tout comme la collecte finale, mais les temps de calculs restent relativement élevés (cf. §3.2, page 136).

Il serait intéressant d'utiliser les reconstructions coûteuses les plus proches de la réalité dans les cas dégénérés et les reconstructions basiques dans les cas simples. Nous n'avons pas étudié ce do-

maine de recherche ; nous n'avons utilisé qu'un type de reconstruction par processus de reconstruction de la solution.

Les résultats fournis dans la section suivante expliciteront notre choix.

3 Résultats.

Dans cette section, nous allons exposer les différents résultats obtenus par les méthodes proposées, résultats en gain visuel et en coûts de calculs. Nous terminerons par déterminer la meilleure solution en terme de rapport de ces deux critères.

3.1 Qualité visuelle.

Dans la majorité des cas, les résultats visuels sont de qualité supérieure à ceux obtenus par l'interpolation finale classique. Quand les liens sont tous établis au niveau des éléments terminaux de la hiérarchie, la qualité est équivalente. A partir de la deuxième itération, la différence de niveau d'établissement a de fortes chances de devenir très importante.

La figure suivante est un gros plan d'un angle d'une pièce rectangulaire. L'interpolation finale fait apparaître le motif du niveau d'établissement des liens (fig. 6.9a). Les deux autres types d'interpolation donnent des résultats équivalents éliminant ces motifs (fig. 6.9b).

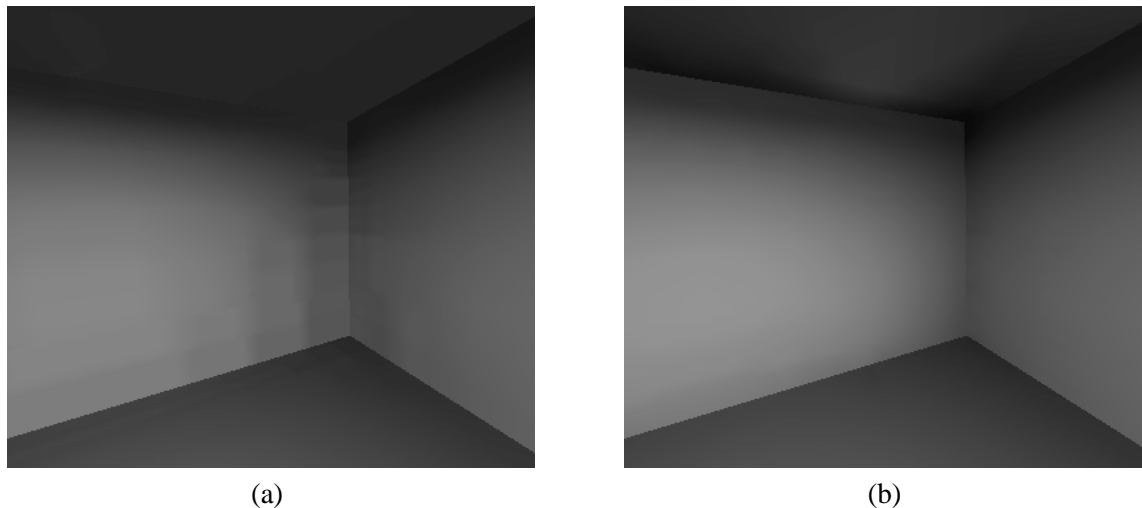


FIG. 6.9: Reconstruction de la fonction de radiosité entre deux murs d'une pièce. (a) interpolation finale. (b) reconstruction par collecte immédiate généralisée ou par collecte par sommets. Les résultats sont visuellement identiques.

3.2 Temps de calcul.

Les temps de calculs sont très dépendants de la scène et du raffinement. Notre exemple est une scène faiblement complexe (6500 polygones) mais fortement illuminée par éclairage indirect. Ce cas est normalement beaucoup plus lent avec nos méthodes de reconstruction que l'interpolation finale car l'éclairage indirect crée des liens à des niveaux élevés de la hiérarchie.

Les différentes méthodes de reconstruction font légèrement varier la radiosité de la solution car les deux méthodes de collecte immédiate généralisée et de collecte par sommet sont plus précises dans les calculs d'irradiance. Le raffinement se base sur la radiosité de la scène pour établir les liens, donc il est aussi légèrement modifié par la méthode de reconstruction. Pour une solution donnée, on ne peut pas utiliser une reconstruction qui intègre la visibilité si le calcul de la solution a été fait sans stocker la visibilité. Les valeurs stockées par lien sont de nature différente.

Comme il n'est pas possible de calculer la reconstruction intégrant la visibilité par une reconstruction ne l'ayant pas stockée au cours du raffinement, nous présentons les résultats par lien. Même si les temps de reconstruction absolus ne peuvent être comparés entre eux, nous les donnerons aussi afin de pouvoir comparer leur ordre de grandeur avec le temps de raffinement (tab. 6.1). Chaque solution a été établie entre 400.000 et 500.000 liens. Le temps de raffinement est de l'ordre de 600 secondes.

Méthode	temps de reconstruction	
	$10^{-6} \cdot s \cdot lien^{-1}$	s
Normale	3.9	1.4
Collecte immédiate	10.7	5.0
Collecte par sommets	83.4	44.2

Méthode avec recalcul de la visibilité

Collecte immédiate	609.6	235.6
Collecte par sommets	2951.0	1421.9

TAB. 6.1: Temps de calcul de la reconstruction de la solution de radiosité en fonction de la méthode utilisée.

Nous pouvons voir, dans cet exemple, une multiplication du temps de calcul légèrement inférieure à 3 pour le cas de collecte immédiate généralisée et de plus de 20 pour la collecte par sommet. Malgré une augmentation importante, les temps restent faibles relativement au temps de raffinement : moins de 1% pour la collecte immédiate généralisée et moins de 10% pour la collecte par sommet.

Le recalcul de la visibilité multiplie les temps par 30 et 60, pouvant dépasser le temps de raffinement. Les calculs de visibilité sont très coûteux en temps de calcul, il est préférable dans le cas de la reconstruction de les stocker durant le processus de raffinement.

Il est à noter que le raffinement utilisé n'est basé que sur de l'échantillonnage, donc les calculs de visibilité ont été réexécutés pour tous les éléments impliqués sans exception*.

3.3 Conclusion.

Les méthodes de reconstruction présentées dans cette section permettent d'éliminer les artefacts induits par l'interpolation finale dus à l'établissement de liens à des niveaux élevés dans la hiérarchie. Les résultats obtenus sont visuellement meilleurs et les coûts de calcul sont très variables. Les deux méthodes ont des résultats similaires mais à des coûts très différents, essentiellement à cause du grand nombre de calculs identiques effectués dans la collecte par point. La collecte immédiate généralisée est donc le meilleur compromis entre qualité de solution et temps de calcul.

L'intégration de visibilité est très lourde en temps de calcul mais la reconstruction peut n'être effectuée qu'une seule fois pour une solution dont le raffinement est déjà réalisé. En intégrant la visibilité après coup, la reconstruction par collecte immédiate est similaire à une collecte par pixel

*L'échantillonnage ne peut pas garantir les classifications VISIBLE et INVISIBLE (cf. §4.4, page 85).

indépendante du point de vue car elle est effectuée par élément de surface terminal. Quand elle est utilisée au cours de la solution, elle est une reconstruction plus fidèle de la fonction de radiosité intégrant l'interpolation.

Conclusion

- *Quel ennui ! C'est toujours la même histoire ! Quand on a fini de construire sa maison, on remarque qu'on a, sans s'en rendre compte, appris en la bâtissant une chose qu'il aurait absolument fallu savoir - avant de commencer. L'éternel et douloureux "trop tard !" - mélancolie de tout ce qui est achevé...*

Friedrich NIETZSCHE

Ce mémoire décrit des travaux concernant la radiosité hiérarchique, technique d'illumination globale basée sur les échanges radiatifs entre éléments de surface purement diffus. L'algorithme calcule l'éclairage d'un environnement en propageant itérativement la lumière ; chaque surface reçoit de l'énergie lumineuse et en réémet une partie. Ceci jusqu'à obtenir une solution stable de la fonction d'éclairage : la *fonction de radiosité*.

Nous avons abordé toutes les étapes de la création de la solution en commençant par les traitement des données en amont, le calcul de la solution en elle-même, puis la phase de reconstruction de cette solution pour obtenir une image. Le calcul de la solution, étape principale déterminant la qualité globale de la solution, repose sur une estimation de la visibilité entre les éléments de la scène. Ce calcul de visibilité est très coûteux et souvent mal estimé. Notre dernière contribution concerne une étude statistique de la visibilité.

Pré-traitement.

Les données en entrée de l'algorithme de radiosité sont une description géométrique complète de la scène. Elles doivent être de préférence une description surfacique par polygones plans car les autres descriptions sont difficilement intégrables dans le calcul de la solution. La radiosité hiérarchique crée une subdivision des surfaces adaptée à l'éclairage local, elles doivent donc être les moins subdivisées possible avant le début de l'algorithme. De plus, un nombre important d'approximations est fait dans

le calcul des échanges radiatifs entres éléments hiérarchiques. Les petits polygones dégénérés sont une source non négligeable des approximations effectuées dans la solution.

Les contraintes géométriques sont donc très fortes quant à la description de la scène. Les modèles réalisés par les logiciels du commerce ne tiennent pas compte de ces contraintes, et les solutions d'éclairage que l'on peut calculer directement avec posent de nombreux problèmes nécessitant fréquemment une intervention humaine au niveau de la géométrie. Nous avons cherché à automatiser une partie de ce pré-traitement des données en remplaçant une partie de la géométrie initiale par une géométrie plus simple et mieux adaptée au calcul d'une solution de radiosité.

Dans l'esprit du *maillage virtuel* créée par l'équipe du LORIA à Nancy, nous avons utilisé des *avatars de surfaces* qui remplacent un ensemble de polygones coplanaires par un unique polygone les englobant. La génération de ces avatars est une tâche fastidieuse exécutée jusqu'alors manuellement. Nous avons décrit trois classes d'algorithmes permettant la génération automatique d'avatars de surfaces plans, ainsi qu'une étude des paramètres dirigeant la création. Finalement, nous avons mis en avant un algorithme particulier avec des paramètres bien définis permettant d'obtenir une solution proche de la solution optimale.

Les avatars de surfaces simplifient la géométrie de la scène et réduisent ainsi les temps de calculs. De plus, le contrôle obtenu sur la géométrie permet de générer des avatars qui limitent les approximations faites sur le calcul du noyau de la fonction de radiosité et augmentent sa qualité.

Le calcul de la solution de radiosité.

Le calcul de l'éclairage d'un environnement se définit par la résolution d'un système d'équations représentant les échanges énergétiques. Les termes principaux de ces équations sont les *facteurs de forme* qui déterminent la quantité d'énergie transitant d'une surface vers une autre en prenant en compte le facteur de visibilité. Le calcul analytique des facteurs de forme est coûteux en temps et son expression n'est pas connue dans les cas de visibilité partielle. Il est donc quasiment toujours une approximation plus ou moins grossière. De plus, si l'on utilise le *regroupement de surfaces*, le "facteur de forme" entre une cellule et un autre élément hiérarchique n'a pas de sens physique. L'ensemble de ces approximations rendent la radiosité instable dans certaines configurations.

La radiosité hiérarchique avec regroupement calcule une solution de l'éclairage en effectuant les calculs d'échanges énergétiques à des niveaux de hiérarchie suffisamment élevés pour réduire les temps de calculs et suffisamment profonds pour que l'erreur commise ne soit pas trop importante. Le choix des niveaux d'établissement des échanges se fait par une heuristique : l'*oracle de raffinement*. Il a aussi la charge de choisir quel type d'approximation de calcul de facteur de forme et de facteur de visibilité utiliser en fonction de la distribution géométrique et énergétique locale à la paire d'éléments traitée. Le nombre de paramètres et le nombre de décisions à prendre sont importants, ce qui rend cette tâche très difficile.

Les quinze années d'existence de l'algorithme de radiosité hiérarchique ont vu plusieurs essais d'oracles utilisant des optiques différentes. Certains se basant sur la quantité d'énergie échangée, sur un encadrement de l'erreur commise, d'autres sur un calcul du gradient de la fonction de radiosité... Chacune de ces heuristiques résout un cas particulier de configuration géométrique et bute sur le même problème : l'intégration de la visibilité. L'oracle de raffinement universel n'a pas encore été découvert.

Pour étudier le raffinement d'une façon systématique, nous avons décrit une structure du processus de raffinement qui le découpe en tâches élémentaires (*outils*) composées entre elles sous la forme d'un *graphe de raffinement*. Nous proposons une classification des outils ainsi qu'une liste des outils les plus utilisés dans le raffinement.

La structure de graphe possède intrinsèquement de nombreux avantages qui permettent une investigation précise notamment grâce à un calcul automatique de statistiques. Après avoir validé le graphe de raffinement sur l'oracle de raffinement le plus utilisé, nous présentons un exemple d'utilisation de cette structure intégrant la visibilité.

Pour finir ce chapitre, nous émettons l'idée d'un *raffinement collaboratif* avec intervention humaine dans les cas où l'oracle n'arrive pas à déterminer correctement la configuration. Ceci permet de centrer la quantité de calculs sur les zones que l'utilisateur juge importantes ainsi que de palier artificiellement aux erreurs produites pendant le calcul de la solution.

Reconstruction.

Une fois l'ensemble des échanges radiatifs calculés au niveau hiérarchique déterminé par le raffineur, il faut conserver la cohérence énergétique de la hiérarchie. Chaque élément doit intégrer l'irradiance qu'il reçoit directement ainsi que celle que ses ascendants lui communiquent, de la même façon, il doit donner son information d'irradiance à ses descendants. Lorsque cette phase, dite de *push/pull*, est terminée l'éclairage de la scène est connu sous la forme d'une fonction par morceaux, où chaque morceau représente localement la radiosité en fonction de la base utilisée. Dans le cas de valeur de radiosité constante par élément, une phase d'interpolation intervient afin d'éliminer les discontinuités de premier ordre. L'ensemble du push/pull et de l'interpolation forme la reconstruction.

La reconstruction telle qu'elle est définie habituellement est bien divisée dans les deux étapes que l'on vient de décrire, de ce fait, l'interpolation se fait au niveau des éléments terminaux. Ceci engendre dans certaines configurations des hautes fréquences dans la solution résultante malgré la continuité effective.

Nous avons proposé une méthode de reconstruction combinant les deux étapes de push/pull et d'interpolation permettant d'éliminer les hautes fréquences désagréables à l'oeil. L'interpolation est effectuée par effet de bord en propageant l'irradiance d'un lien immédiatement à ses fils.

Visibilité.

L'estimation du facteur de visibilité entre deux éléments est une tâche difficile et longue à réaliser. C'est souvent un des goulots d'étranglement des algorithmes qui doivent l'évaluer. La radiosité n'échappe pas à cette règle car non seulement le noyau de l'échange radiatif contient une évaluation du facteur de visibilité mais la connaissance de la classification de la visibilité est fortement utile pour la décision de l'oracle de raffinement.

Les techniques d'échantillonnage ne sont pas fiables dans des conditions où les obstacles sont petits ou très denses par rapport à la taille du *pinceau de visibilité* (volume d'évaluation de la visibilité). Les volumes de vision sont un peu plus coûteux et n'estiment pas le facteur de visibilité. Les estimations basées sur le matériel graphique sont encore trop coûteuses mais sont très prometteuses. Aucun algorithme ne sait déterminer précisément l'invisibilité totale, ce qui permettrait pourtant de réduire fortement le raffinement d'une solution de radiosité.

Nous avons étendu les volumes de vision afin de permettre de déterminer certains cas d'invisibilité totale grâce à la combinaison d'une intersection volumique et d'un échantillonnage bien choisi. Nous avons présenté, dans l'optique décrite par le créateur des volumes de vision, une génération optimisée des *volumes de vision étendus*.

Nous avons testé une optimisation des intersections par échantillonnage basée sur la cohérence spatiale des obstacles en plus de la cohérence spatiale des éléments pour lesquels la visibilité est

évaluée. Celle-ci requiert une hiérarchie de la scène afin de pouvoir la traverser à partir du dernier obstacle intersecté. La mise en oeuvre n'est pas aussi aisée qu'un cache d'intersection et les résultats sont presque identiques. Ainsi nous ne recommandons cette méthode que dans des cas particuliers de géométrie très dense.

Pour classer la visibilité il existe plusieurs types d'outils, chacun garantissant plus ou moins les réponses qu'il peut donner. Le choix de l'outil est important pour le rapport entre le temps de calcul dépensé et la certitude de la réponse. Nous avons effectué une étude statistique de la visibilité d'un pinceau quelconque dans un environnement chaotique d'obstacles variant en fonction de paramètres de taille d'obstacles et de densité. Nous avons établi la corrélation entre la théorie et un calcul en masse sur une scène créée en fonction de ces mêmes paramètres. Enfin, nous avons créé un outil de prédiction basé sur ces statistiques afin de choisir dans certains cas, l'outil de classification de visibilité probablement le plus adapté à la configuration des obstacles.

Pour conclure.

La radiosité hiérarchique est une méthode datant d'une quinzaine d'années qui a connu un essor très important dès sa création. Actuellement, les recherches se tournent plutôt vers d'autres algorithmes d'illumination globale pour plusieurs raisons :

- Difficulté de mise en oeuvre : l'algorithme a beaucoup de phases différentes (calcul du facteur de forme, calcul du facteur de visibilité, transfert d'énergie, push/pull, reconstruction, etc.) et les précis sont rares. Les deux principaux existants sont relativement anciens [CW93, Ash94, SP94] et, en particulier, ne parlent pas en détail du regroupement de surfaces.
- Instabilité : les différentes approximations faites à tous les niveaux et l'absence d'heuristique bien définie contrôlant celles-ci font que les calculs n'ont pas l'exactitude que les premiers algorithmes analytiques pouvaient avoir.
- Dépendance au modèle géométrique : les approximations d'évaluation du facteur de forme dépendent fréquemment de la forme des éléments hiérarchiques mis en situation. La géométrie initiale doit être la moins subdivisée possible et éviter de comporter des polygones dégénérés, ce qui limite l'utilisation de logiciels de modélisation commerciaux.
- Raffinement : la majeure partie des problèmes posés lors des simplifications et approximations faites par la radiosité hiérarchique a été reportée sur l'oracle de raffinement. Il lui incombe la lourde tâche de choisir tous les nombreux paramètres de l'algorithme : choix de l'estimateur de calcul de facteur de forme, choix de l'estimateur de calcul de visibilité, choix du niveau d'établissement et des éléments des échanges énergétiques. La réalisation d'un tel raffineur est donc très difficile.
- Contrôle de la solution : actuellement, et de par l'absence d'oracle bien défini, le contrôle de l'utilisateur sur la solution passe par une connaissance forte de l'algorithme et de ses paramètres. Les grandeurs utilisées sont fréquemment relatives à la scène et même relatives à une configuration locale. Un jeu de paramètre pour une scène complète est rarement suffisant pour un environnement complexe.

La radiosité hiérarchique a progressé sur la relègue d'une bonne partie des problèmes en attendant que le raffinement soit clairement solutionné. Par exemple, le regroupement de surfaces permet de réduire la complexité de la radiosité en supposant que l'oracle de raffinement évaluera correctement les approximations engendrées durant le calcul de la solution. Hélas, le choix de l'oracle repose sur une estimation de la visibilité qui, actuellement, n'a pas de solution générique convaincante en des temps raisonnables. Les approximations engendrées par les différentes simplifications/évolutions de l'algo-

rithme semblent trop importantes pour qu'une heuristique puisse prendre une décision suffisamment juste à chaque fois.

Nous avons offert quelques solutions sur les différents points négatifs de la radiosité et créé des outils permettant l'analyse des problèmes de manière plus systématique. Malgré les contributions apportées, nous n'avons, à grands regrets, pas réussi à créer dans son ensemble un oracle de raffinement universel, notamment à cause de la visibilité.

Dans la même optique que les meilleurs joueurs d'échec de la planète arrivent encore à battre un ordinateur, l'esprit humain arrive à déterminer beaucoup plus précisément qu'une heuristique à quel niveau les échanges énergétiques doivent se faire. En attendant que la puissance des ordinateurs vienne aider à la résolution de calcul de solutions sur ce qui est actuellement des scènes complexes, nous proposons d'introduire un facteur humain. L'algorithme de radiosité n'est alors plus un algorithme de simulation automatique mais un système de simulation d'éclairage assisté par ordinateur.

Pistes de recherche

Quand y'en a plus, y'en a encore.

expression populaire française

La radiosité est un algorithme complexe composé de plusieurs parties : algorithme de regroupement de surfaces, résolution du système d'équations de luminance, estimation du facteur de forme et du facteur de visibilité, reconstruction de la solution, algorithme de push/pull, etc. Malgré les nombreuses recherches effectuées et les solutions apportées sur des points particuliers, il reste beaucoup de problèmes à résoudre autant ponctuellement qu'au niveau de l'interaction entre les différentes étapes.

Nous proposons ici quelques pistes de recherches que nous n'avons pas eu le temps d'aborder ou pour lesquels notre investigation n'est pas complètement aboutie.

Évaluation qualitative d'un raffinement.

Lorsque l'on n'a pas de référence, il est difficile d'émettre un critère de qualité sur un objet. Le problème classique en imagerie de synthèse est de juger la qualité d'une image créée à base d'une description mathématique complètement artificielle. Comment définir une *description de la qualité* de l'image générée ?

L'humain peut donner une critique dans une partie des situations, notamment lorsque il a une grande connaissance de la classe d'objet à obtenir. Si l'image résultante doit être un visage, l'ensemble des caractéristiques de l'image correspondantes à la notion de visage vont être comparées à notre représentation schématique interne de visage. Notre représentation schématique est basée sur l'apprentissage par l'exemple et la mémorisation. Les structures de données informatiques correspondantes sont les systèmes à base de connaissance.

Juger la qualité d'un raffinement est un problème similaire à celui de juger la qualité d'une image. Le problème est toutefois plus compliqué car les données ne sont plus un échantillonnage uniforme de pixels (radiance) en 2D mais un échantillonnage quelconque de fonctions de radiosit  en 3D.

La piste d'un syst me   base de connaissance semble un peu trop ambitieuse. Par contre, il est possible d' valuer des crit res de plus bas niveau tels que les discontinuit s, les diff rences de niveaux hi rarchiques d' tablissement des  changes  nerg tiques entre  l ments contigus, etc.

Mais en utilisant de tels crit res, se posent les m mes probl mes que pour le jugement qualitatif d'images et, entre autres, une instabilit  par rapport   des crit res math matiques  loign s de nos crit res perceptifs. La connaissance des crit res perceptifs humains est,   ce jour, insuffisante pour esp rer en tirer un crit re d' valuation de la qualit  d'une solution.

Cellules sp cialis es.

Les cellules cr ees par un algorithme de regroupement de surfaces sont, dans la majeure partie des cas, anisotropes. L'hypoth se principale de la radiosit  est de travailler sur un ensemble de surfaces lambertiennes donc isotropes. L'implantation basique des cellules dans un algorithme de radiosit  hi rarchique produit donc des approximations tr s importantes, difficilement  valuables par l'oracle de raffinement.

Les cellules, parce qu'elles ne rentrent pas exactement dans le cadre de la radiosit , doivent contenir plus d'information que les  l ments de surface. Sans aller jusqu'  une repr sentation compl te de la BRDF d'une cellule, certaines donn es simples peuvent parfois  tre repr sentatives du contenu de la cellule. Par exemple conna tre le c ne des normales pour des surfaces dirig es plus ou moins dans la m me direction, savoir si les surfaces sont connexes, avoir une estimation du facteur de visibilit  interne dans des directions privil gi s, avoir une repr sentation de la BRDF simplifi e si elle est simplifiable (*avatars 3D*), conna tre la densit  et la distribution des surfaces dans certaines directions ou encore simplement conna tre une valeur d'homog nit  d' mission directionnelle ou non.

Cet ensemble de connaissances suppl mentaires permettrait, dans quelques cas, de limiter les approximations faites en utilisant un estimateur plus adapt  aux cellules mises en jeu. Les informations peuvent  tre autant de nature g om trique,  nerg tique ou ayant rapport   la visibilit .

Deux difficult s apparaissent : la premi re est de trouver les informations pertinentes   partir de la description de la sc ne. Sachant que l'algorithme de regroupement de surfaces, qui doit simplement d terminer un regroupement g om trique, n'a pas de solution optimale connue, la d termination de crit res plus complexes n'est pas  vidente.

Le second probl me r side dans l'utilisation de ces informations. Quand bien m me elles sont suppos es aider   la d cision du raffinement, la quantit  d'informations est accrue. L'oracle de raffinement doit alors chercher l'information correspondante   la configuration g om trique dans un ensemble plus vaste. De plus, cette information n'est pas forc ment globale   la sc ne et ne pourra pas, dans ce cas,  tre calcul e en pr -traitement.

Reclassification des sources.

On parle souvent d' clairage direct et d' clairage indirect. La plupart des ombres notables sont dues   l' clairage direct parce que plus puissant. Il est possible de calculer l' clairage direct avec un algorithme capable de cr er des ombres douces ou dures facilement et sans subdivision de la g om trie tel qu'un algorithme de tampon de profondeur distribu  [Kel97]. La solution de radiosit 

est alors calculée sans les sources primaires. Un rendu à base de textures et de transparence permettrait d'afficher toujours en temps réel.

Des ombres sont aussi générées par éclairage indirect dans certains environnements. Il est alors imaginable de classer certaines sources comme primaires et secondaires et d'utiliser à chaque itération du calcul de la solution de radiosité deux algorithmes différents de calcul de l'éclairage. Le push/pull serait alors la phase de recombinaison des données en une représentation unique (sûrement la géométrie).

Les solutions utilisant deux méthodes différentes posent le problème de la jonction entre les deux algorithmes, ainsi que du choix du domaine d'application de chaque algorithme. De plus, le rendu à base de textures nécessite une heuristique de leur caractérisation (taille, position, définition) qui n'est pas évidente.

Stochastique guidé.

La radiosité stochastique est une piste de recherches intéressante qui permet de combiner la simplicité de l'algorithme stochastique à la création de scènes éclairées indépendamment du point de vue. Le fait que les surfaces soient lambertiennes empêche l'utilisation d'une fonction d'importance et limite la rapidité de la convergence de l'algorithme.

Une solution rapide de radiosité hiérarchique pourrait être utilisée comme fonction d'importance afin d'obtenir une convergence plus rapide.

Raffinement semi-automatique.

Nous avons présenté dans ce mémoire une première approche de raffinement automatique où l'utilisateur peut intervenir manuellement dans la solution afin de demander une précision plus importante dans certaines zones d'éclairage.

Plusieurs types d'aides peuvent être imaginées pour améliorer la collaboration entre l'algorithme et l'utilisateur. Par exemple, l'utilisateur pourrait peindre directement sur la géométrie pour adoucir des discontinuités pendant que l'ordinateur garantit la conservation de la cohérence hiérarchique. L'utilisateur peut aussi décider d'obtenir des ombres précises sur un objet. Tout ceci sans création d'erreurs "humaines" grâce au contrôle de l'algorithme.

Un travail important relevant de l'interfaçage homme-machine peut être exécuté afin d'offrir un outil où la machine serait un outil d'aide à la simulation d'éclairage.

Repérage automatique des discontinuités.

Un des artefacts visuel les plus fréquents est la présence de discontinuités de la fonction de radiosité dues à la géométrie. Ces discontinuités peuvent être repérées (sûrement à un coût important) grâce aux niveaux d'établissement des liens qui sont dans ce cas relativement éloignés.

Tout comme la hiérarchie de surfaces est contrainte afin d'éviter des artefacts de visualisation, on pourrait contraindre la différence de niveau d'établissement des liens entre éléments de hiérarchie contigus. Une difficulté est que les contraintes d'établissement sont définies par émetteur, ce qui peut poser des problèmes à caractériser.

Le repérage automatique des discontinuités après calcul de la solution peut également être une des aides fournies à l'utilisateur par un algorithme semi-automatique.

Pré-raffinement.

La première étape de raffinement part de la cellule racine de la hiérarchie des éléments de la scène et subdivise fortement jusqu'au surfaces dans le cas d'éclairage direct. Un temps non négligeable est passé dans les tests du haut de la hiérarchie et sont réexécutés (souvent similairement) à chacun des niveaux hiérarchiques. L'évaluation de la visibilité étant un des points les plus long.

Le raffinement peut être vu comme une suite convergente de fonctions de radiosité par morceaux. Pour améliorer la vitesse de convergence d'une suite, nous pouvons modifier la valeur initiale de celle-ci en la choisissant plus proche de la valeur de convergence.

Nous pouvons imaginer une évaluation de la visibilité globale en pré-traitement tel qu'introduit par Mateu Sbert [Sbe97]. Cette étape permettrait d'éliminer certaines interactions classifiées directement INVISIBLE et, au contraire, d'en établir grâce à une classification VISIBLE. Le même principe doit-être applicable au niveau de l'énergie de la scène.

Cette approche reste pour l'instant très floue, et doit faire l'objet du étude un peu plus approfondie afin de déterminer si c'est envisageable ou non.

Transformation espace sphérique - espace cartésien

La transformation des coordonnées cartésiennes en coordonnées sphériques est une transformation simple et fréquemment utilisée en synthèse d'image, elle requiert toutefois une certaine attention au niveau des discontinuités de l'espace destination. De plus la transformation présentée ici est légèrement modifiée par rapport à la transformation classique, car nous transformons 6 coordonnées (un point p et un vecteur \vec{n}) en 3 coordonnées ρ , Θ et Φ représentatif du même plan.

1 Transformation point - point

Transformations des coordonnées cartésiennes d'un point M quelconque (x, y, z) en coordonnées sphériques (ρ, Θ, Φ) et inversement. Dans l'espace sphérique, la coordonnée ρ représente la distance du point M à l'origine, et Θ, Φ la direction longitude, latitude du point à partir de l'origine (fig. A.1).

$$(x, y, z) \in \mathbb{R}^3$$
$$(\rho, \Theta, \Phi) \in \mathbb{R} \times [-\pi; \pi] \times [0; \pi]$$

1.1 Espace sphérique \rightarrow espace cartésien

$$\begin{cases} x = \rho \cos \Theta \sin \Phi \\ y = \rho \sin \Theta \sin \Phi \\ z = \rho \cos \Phi \end{cases}$$

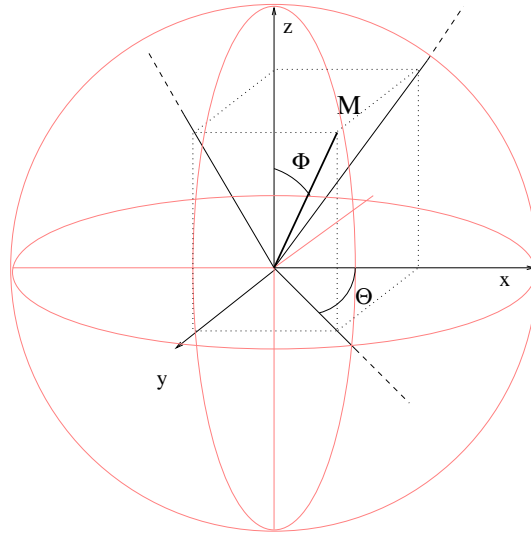


FIG. A.1: Transformations entre espace sphérique et espace cartésien.

1.2 Espace cartésien \rightarrow espace sphérique

$$\begin{cases} \rho = \sqrt{x^2 + y^2 + z^2} = \|\vec{OM}\| \\ \Theta = \arctan \frac{y}{x} \\ \Phi = \arctan \frac{\sqrt{x^2 + y^2}}{z} = \arccos \frac{z}{\rho} \end{cases}$$

2 Transformation (point,vecteur) - point

Le but de cette transformation est de définir un plan en coordonnées sphériques. Pour tout plan défini par un point M et une normale \vec{n} , il existe un triplet (ρ, Θ, Φ) unique. Cette transformation réduit 6 coordonnées en 3, il y a une perte d'information, les informations perdues sont en l'occurrence les coordonnées du point appartenant au plan. La seule représentation qui nous intéresse étant le plan en lui-même, la position de ce point n'est pas discriminante.

2.1 Espace sphérique \rightarrow espace cartésien

Nous devons à partir de ρ, Θ, Φ pour reconstruire la normale ainsi qu'un point appartenant au plan. Le point appartenant au plan est arbitrairement choisi comme le point le plus proche de l'origine (H).

$$\begin{cases} H_x = \rho \cos \Theta \sin \Phi \\ H_y = \rho \sin \Theta \sin \Phi \\ H_z = \rho \cos \Phi \end{cases} \quad \begin{cases} \vec{n}_x = \cos \Theta \sin \Phi \\ \vec{n}_y = \sin \Theta \sin \Phi \\ \vec{n}_z = \cos \Phi \end{cases}$$

2.2 Espace cartésien \rightarrow espace sphérique

Il y a beaucoup de discontinuités et de cas particuliers invoqués dans la transformation de l'espace cartésien dans l'espace sphérique, de plus la fonction arccos n'est bijective que dans une demie révolution, il faut utiliser ses composantes pour connaître son signe. Voici les différents cas décomposés :

$x = 0$	$y = 0$	$z = 0$	$\rho = 0$	$\Theta = 0$	$\Phi = 0$
$x = 0$	$y = 0$	$z > 0$	$\rho = \ \vec{OM}\ $	$\Theta = 0$	$\Phi = 0$
$x = 0$	$y = 0$	$z < 0$	$\rho = \ \vec{OM}\ $	$\Theta = 0$	$\Phi = \pi$
$x > 0$	$y = 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = 0$	$\Phi = \frac{\pi}{2}$
$x < 0$	$y = 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = \pi$	$\Phi = \frac{\pi}{2}$
$x = 0$	$y > 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = \frac{\pi}{2}$	$\Phi = \frac{\pi}{2}$
$x = 0$	$y < 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = -\frac{\pi}{2}$	$\Phi = \frac{\pi}{2}$
$x = 0$	$y > 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = \frac{\pi}{2}$	$\Phi = \arccos \vec{n}_z$
$x = 0$	$y < 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = -\frac{\pi}{2}$	$\Phi = \arccos \vec{n}_z$
$x > 0$	$y = 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = \pi$	$\Phi = \arccos \vec{n}_z$
$x < 0$	$y = 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = 0$	$\Phi = \arccos \vec{n}_z$
$x < 0$	$y < 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = \arctan \frac{\vec{n}_y}{\vec{n}_x} - \pi$	$\Phi = \frac{\pi}{2}$
$x < 0$	$y > 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = \arctan \frac{\vec{n}_y}{\vec{n}_x} + \pi$	$\Phi = \frac{\pi}{2}$
$x > 0$	$y \neq 0$	$z = 0$	$\rho = \ \vec{OM}\ $	$\Theta = \arctan \frac{\vec{n}_y}{\vec{n}_x}$	$\Phi = \frac{\pi}{2}$
$x > 0$	$y \neq 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = \arctan \frac{\vec{n}_y}{\vec{n}_x}$	$\Phi = \arccos \vec{n}_z$
$x < 0$	$y > 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = \arctan \frac{\vec{n}_y}{\vec{n}_x} + \pi$	$\Phi = \arccos \vec{n}_z$
$x < 0$	$y < 0$	$z \neq 0$	$\rho = \ \vec{OM}\ $	$\Theta = \arctan \frac{\vec{n}_y}{\vec{n}_x} - \pi$	$\Phi = \arccos \vec{n}_z$

Remarque : Si H est la projection de l'origine sur le plan défini par le point M et la normale \vec{n} , le vecteur \vec{OH} est colinéaire au vecteur \vec{n} seulement quand \vec{OH} n'est pas nul. À cause de cette singularité, le point H est le vecteur \vec{n} ne sont pas redondants. Il est nécessaire d'utiliser les deux paramètres M et \vec{n}

TABLE DES MATIÈRES

Introduction	1
1 Le rendu en synthèse d'image	5
1 Lumière et grandeurs lumineuses.	6
1.1 Qu'est-ce que la lumière ?	6
1.2 L'oeil humain.	7
1.3 Radiométrie et photométrie.	9
1.4 Unités de mesure	10
1.5 Ordres de grandeurs réels.	13
2 Les algorithmes de rendu.	14
2.1 Rendu photoréaliste.	14
2.2 Théorie.	15
2.3 Couleur.	15
2.4 Fonction de réflectance bidirectionnelle.	16
3 Modèles d'éclairage local.	18
3.1 Rendu lambertien.	18
3.2 Modèle de Phong.	19
4 Modèles d'éclairage global.	19
4.1 Chemins lumineux.	20
4.2 Tracer de rayons.	20
4.3 Tracer de chemins.	21
4.4 Tracer de photons.	21
4.5 Lancer de rayon bidirectionnel.	21
4.6 Radiosité.	22
4.7 Méthodes hybrides.	22
2 L'algorithme de radiosité	25
1 L'échange énergétique.	25
1.1 L'équation de luminance.	25
1.2 Facteurs de forme	27

2	Les algorithmes de radiosité.	30
2.1	Radiosité de Galerkin	31
2.2	Résolution matricielle	31
2.3	Radiosité progressive	31
2.4	Radiosité Hiérarchique	33
2.5	Regroupement de surfaces	35
2.6	Radiosité par ondelettes.	36
3	Variations sur la radiosité.	37
3.1	Réduction de mémoire.	37
3.2	Parallélisation.	37
3.3	Radiosité dynamique.	38
3.4	Création d'images.	39
3.5	Contrôle d'erreur.	39
4	Conclusion	39
3	Avatars de surfaces	41
1	Radiosité et modèles géométriques.	42
1.1	Génération de maillages.	42
1.2	Approximation du facteur de forme.	42
2	Avatars de surfaces.	43
2.1	Avatars de surfaces plans.	44
2.2	Création de la description géométrique.	48
3	Génération automatique d'avatars de surfaces plans.	50
3.1	Réduction de la complexité.	50
3.2	Critères de génération.	50
4	Regroupement en surfaces quasi-coplanaires.	51
4.1	Opérateur de coplanarité.	51
4.2	Algorithme glouton.	52
4.3	Regroupement par hiérarchie.	52
4.4	Regroupement par espace dual.	53
5	Génération d'avatars à partir de surfaces coplanaires.	57
5.1	Paramètres.	57
5.2	Regroupements existants.	58
5.3	Regroupement balancé.	58
6	Résultats.	59
6.1	Résultats visuels.	59
6.2	Temps de création et de solution.	60
6.3	Paramètres.	63
7	Conclusion.	67
4	Le raffinement	69
1	Le raffinement progressif.	70
1.1	Raisons de subdivision.	70
1.2	Oracle de raffinement.	70
2	Outils de raffinement.	71
2.1	Outils d'interrogation.	72
2.2	Outils d'action.	73
3	Structure arborescente de l'oracle de raffinement.	74

3.1	Gestionnaire de données.	75
3.2	Paramétrisation.	75
3.3	Avantages de la structure utilisée.	76
3.4	Validation et temps d'exécution.	76
4	Outils.	78
4.1	Outils géométriques.	78
4.2	Outils énergétiques.	81
4.3	Outils d'action.	82
4.4	Outils de visibilité.	85
5	Usage de la structure.	86
5.1	Oracle initial.	87
5.2	Échantillonnage adaptatif.	87
5.3	Visibilité gérée par volumes de vision étendus.	89
5.4	Visibilité combinée.	89
5.5	Conclusion.	90
6	Raffinement collaboratif.	91
6.1	Détermination des liens.	92
6.2	Exemple d'amélioration d'une ombre.	93
7	Bilan.	94
5	Visibilité	97
1	Classification de visibilité.	98
1.1	Classes.	98
1.2	Volumes de vision.	99
1.3	Volumes de vision étendus.	103
2	Techniques d'accélération.	104
2.1	Motivation.	104
2.2	Lancer de rayons.	106
3	Volumes de vision et raffinement.	110
3.1	Liste de volumes de vision.	111
3.2	Résultats.	112
4	Statistiques de visibilité.	113
4.1	Courbes de référence.	113
4.2	Pinceau infinitésimal.	115
4.3	Pinceau à section constante.	115
4.4	Pinceau à section variable.	119
4.5	Application à un cas concret.	121
4.6	Conclusion.	124
5	Bilan.	125
6	Reconstruction	127
1	Reconstruction de la solution.	127
1.1	Collecte immédiate.	128
1.2	Interpolation finale.	128
1.3	Collecte par pixel.	130
2	Améliorations de la reconstruction.	131
2.1	Interpolation hiérarchique.	131
2.2	Collecte immédiate généralisée.	133

2.3	Collecte par sommet.	134
2.4	Liens et visibilité.	135
3	Résultats.	136
3.1	Qualité visuelle.	136
3.2	Temps de calcul.	136
3.3	Conclusion.	137
Conclusion		139
Pistes de recherche		145
A Transformation espace sphérique - espace cartésien		149
1	Transformation point - point	149
1.1	Espace sphérique \rightarrow espace cartésien	149
1.2	Espace cartésien \rightarrow espace sphérique	150
2	Transformation (point,vecteur) - point	150
2.1	Espace sphérique \rightarrow espace cartésien	150
2.2	Espace cartésien \rightarrow espace sphérique	150
Table des matières		153
Table des figures		157
Bibliographie		165

TABLE DES FIGURES

1.1	Décomposition d'une onde électromagnétique en deux ondes perpendiculaires.	7
1.2	Section d'un globe oculaire humain.	8
1.3	Répartition des cônes et bâtonnets en fonction de l'angle à la fovéa.	8
1.4	Fonction de perception de l'intensité lumineuse de l'oeil humain moyen $V(\lambda)$ et $V'(\lambda)$ respectivement pour les domaines photopiques et scotopiques.	10
1.5	Spectre des rayonnements électromagnétiques.	10
1.6	Différents rendus pour une même géométrie.	14
1.7	BRDF diffuse caractérisant les surfaces lambertiennes.	17
1.8	BRDF à deux composantes, une diffuse et une spéculaire.	17
1.9	Vecteurs utilisés dans le modèle d'éclairage lambertien du modèle d'éclairage de Phong.	19
1.10	Rayons lancés à partir de l'oeil pour calculer récursivement la radiance au pixel désiré.	20
1.11	Rayons lancés simultanément à partir de l'oeil et la source pour calculer récursivement la radiance au pixel désiré.	22
2.1	Géométrie du transfert énergétique.	27
2.2	Facteur de forme entre une surface élémentaire et un polygone.	28
2.3	Facteur de forme entre une surface élémentaire et un disque.	28
2.4	Facteur de forme entre une surface élémentaire et une sphère.	29
2.5	L'hémicube : calcul discret du facteur de forme d'un point vers l'environnement visible de ce point.	30
2.6	Résolution de l'équation de radiosité par les méthodes de Southwell (a) et Gauss-Seidel (b).	32
2.7	Exemple de subdivision localisée dans les zones d'intérêts.	33
2.8	Représentation d'une fonction de radiosité à différents niveaux hiérarchiques.	34
2.9	Push-Pull : Conservation de la cohérence énergétique dans la hiérarchie. (a) Diffusion de l'irradiance du haut de la hiérarchie jusqu'aux feuilles. (b) Moyenne des valeurs d'énergie lumineuse des feuilles jusqu'au sommet.	35
3.1	Triangulation d'une arche par deux logiciels courants de modélisation (a) blender (b) 3D Studio Max.	42

3.2	Approximations des facteurs de formes dans un cas de maillage dégénéré. (a) Maillage de la scène. (b) Artéfacts visuels engendrés.	43
3.3	Création de groupes de surfaces ayant une connaissance supérieure aux groupes de surfaces classiques. (a) Pas de connaissance à priori. (b) Connaissance de coplanarité.	44
3.4	Échantillonnage d'un polygone quelconque par échantillonnage de son polygone englobant.	46
3.5	Visualisation des avatars en utilisant les pochoirs d'Open GL. (a) Géométrie initiale des avatars. (b) Ajout de pochoirs afin d'éliminer les zones invisibles.	47
3.6	Artéfacts de visualisation générés par l'utilisation des pochoirs. (a) Avatars générés sur une sphère. (b) visualisation par pochoirs.	47
3.7	Génération de l'enveloppe convexe rectangulaire d'aire minimale en deux étapes : (a) trouver le meilleur candidat par échantillonnage, (b) affiner la solution par dichotomie.	49
3.8	Découpe du problème de génération automatique d'avatars en sous-problèmes.	50
3.9	Critère de coplanarité entre surfaces dépendant de l'angle α et de la distance h	51
3.10	Création d'avatar par un algorithme glouton	52
3.11	Création d'avatars en utilisant la hiérarchie.	53
3.12	Création d'avatar par l'utilisation d'un espace dual.	54
3.13	Dépendance du paramètre de distance (ρ) à l'origine en coordonnées polaires.	54
3.14	Courbe d'interpolation linéaire bornée pour la valeur des paramètres.	58
3.15	Exemple d'amélioration visuelle induite par l'utilisation des avatars de surfaces.	60
3.16	Niveaux hiérarchiques d'établissement des liens. (a) Sans avatars. (b) Avec avatars.	61
3.17	Suppression de discontinuités entre polygones jointifs par l'utilisation des avatars de surfaces. (a) Solution sans avatars. (b) Solution avec avatars.	61
3.18	Scène de rue comportant beaucoup de surfaces coplanaires.	63
3.19	Arbre dont les feuilles forment un ensemble de surfaces faiblement coplanaires.	64
3.20	Scènes ALBISIA comportant un arbre à deux échelles différentes. (a) Taille importante (de l'ordre du mètre). (b) Taille réduite (de l'ordre du centimètre).	64
3.21	Non corrélation entre régularité et taux de remplissage de l'avatar sur trois différentes scènes. (a) RUE. (b & c) ALBISIA petite et grande taille.	64
3.22	Variation de la génération d'avatars en fonction du paramètre de taux de remplissage pour la scène RUE	66
3.23	Variation de la génération d'avatars en fonction du paramètre de taux de remplissage pour la scène ALBISIA (petite taille)	67
3.24	Variation de la génération d'avatars en fonction du paramètre de taux de remplissage pour la scène ALBISIA (grande taille)	68
4.1	Structuration du processus de raffinement en discussion entre un ensemble d'outils et l'oracle de raffinement.	71
4.2	Exemple de cas de répartition de l'énergie non uniforme à déterminer par un outil énergétique.	72
4.3	Structure arborescente du graphe de décision.	74
4.4	Graphe de décision du critère BF.	77
4.5	Estimation de l'angle solide sous-tendu par deux éléments dont leurs projections sont considérées circulaires.	79
4.6	Solution de radiosité utilisant le critère BF (a) sur la scène des deux bureaux (b). Le lien transitant d'une pièce vers la seconde n'est pas subdivisé.	88

4.7	Sur-raffinement de la solution dans les zones de visibilité nulle ou partielle avec l'utilisation d'un outil de classification par échantillonnage.	88
4.8	Solution de radiosité utilisant une classification de visibilité par volumes de vision (a). Les zones complètement occultées pour l'éclairage direct n'ont pas été subdivisées (b).	89
4.9	Solution de radiosité utilisant une classification de visibilité par un mélange de volumes de vision et d'échantillonnage (a). Le maillage est bien adaptée aux variations de la fonction de radiosité (b).	91
4.10	Raffinement initial où les ombres des objets principaux ne sont pas bien définies.	93
4.11	Sélection d'un ensemble de récepteurs et d'un obstacles afin d'améliorer la solution de radiosité des liens étant impliqués avec eux.	94
4.12	Raffinement précis des ombres dans les zones déterminées par l'utilisateur.	94
5.1	Enveloppe convexe et volume de vision entre deux éléments. (a) Enveloppe convexe. (b) Volume de vision appuyé sur des boîtes englobantes alignées avec les axes. (c) Frontières des demis-espaces et boîte englobante au plus près.	99
5.2	Faces Inutiles dans un volumes de vision. Ici P_1 est redondante avec la boîte englobante.	100
5.3	Exemple de création d'un volume de vision en 3 dimensions.	101
5.4	Différentes configurations géométriques de classification de visibilité par volume de vision.	103
5.5	Scène de test des tests de visibilité composée de multiples petits objets ainsi que d'un grand obstacle central.	106
5.6	Parcours hiérarchique (a) et feuille-feuille (b) pour le calcul d'intersection de rayon (réponse binaire). A : dernier obstacle, B : obstacle actuel.	108
5.7	Scènes de tests de l'algorithme de parcours inverse. (a) scène de l' <i>albisia</i> ayant une forte cohérence géométrique des obstacles. (b) scène du <i>labyrinthe</i> possédant peu d'obstacles de taille importante.	109
5.8	Inclusion des volumes de visions entre un niveau hiérarchique de raffinement et les volumes inférieurs engendrés.	110
5.9	Table de hachage et table des obstacles potentiels pour l'optimisation d'intersections avec des volumes de vision.	112
5.10	Protocole de test utilisé permettant d'isoler les paramètres de taille et nombre d'obstacles, ainsi que la taille du pinceau (=taille des astres).	114
5.11	Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau cylindrique. (page 1/2)	117
5.12	Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau cylindrique. (page 2/2)	118
5.13	Géométrie de la tangence utilisée entre un disque représentatif d'une sphère et d'un cône (représentation en coupe).	120
5.14	Volume d'occultation totale dans un pinceau conique.	120
5.15	Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau conique. (page 1/2)	122
5.16	Courbes de référence calculées comparées aux courbes théoriques estimées pour un pinceau conique. (page 2/2)	123
6.1	Conservation de la cohérence de la hiérarchie et reconstruction par interpolation finale. (a) niveau d'établissement des liens. (b) collecte. (c) push. (d) pull. (e) interpolation finale.	129

6.2	Variation de l'interpolation finale en fonction de la hauteur d'établissement des liens dans la hiérarchie. (a) établissement des liens au niveau de la géométrie. (b) solution reconstruite par interpolation finale de (a). (c) établissement des liens à des niveaux hiérarchiques supérieurs à la géométrie. (d) solution reconstruite par interpolation finale de (c).	130
6.3	Schématisation d'un parcours hiérarchique et des différents traitements applicables.	130
6.4	Description de la collecte et du push/pull de l'interpolation finale.	131
6.5	Noyau d'interpolation symétrique permettant de conserver les deux propriétés de valeur centrale et de conservation de l'énergie. (a) 2 dimensions. (b) 3 dimensions.	133
6.6	Reconstruction en 2 dimensions par interpolation d'une fonction constante dont les valeurs sont définies à des niveaux hiérarchiques différents.	133
6.7	Description de la collecte et du push/pull de la collecte immédiate généralisée.	134
6.8	Description de la collecte et du push/pull de la collecte par sommet.	135
6.9	Reconstruction de la fonction de radiosit� entre deux murs d'une pi�ce. (a) interpolation finale. (b) reconstruction par collecte imm�diate g�n�ralis�e ou par collecte par sommets. Les r�sultats sont visuellement identiques.	136
A.1	Transformations entre espace sph�rique et espace cart�sien.	150

LISTE DES TABLEAUX

3.1	Calcul d'une solution de radiosité sans avatars	62
3.2	Calcul d'une simulation de radiosité avec avatars de surfaces paramétrée par un taux de remplissage élevé.	62
3.3	Calcul d'une simulation de radiosité avec avatars de surfaces paramétrée par un taux de remplissage étendu.	62
4.1	Comparaison des temps de calculs des versions itératives, récursives et normales d'un raffineur utilisant le critère BF.	78
4.2	Statistiques du raffinement pour le raffineur utilisant une le critère BF et une classification de visibilité par échantillonnage adaptatif.	89
4.3	Statistiques du raffinement pour le raffineur utilisant une classification de la visibilité uniquement par volumes de vision.	90
4.4	Statistiques du raffinement pour le raffineur utilisant une classification de la visibilité uniquement par volumes de vision.	91
4.5	Temps de mise à jour d'ombres à l'aide d'un raffineur collaboratif.	95
5.1	Temps de raffinement utilisé par les outils de visibilité pour un raffineur basé sur un critère BF dont la visibilité est évaluée par échantillonnage.	106
5.2	Temps de raffinement utilisé par les outils de visibilité pour un raffineur basé sur un critère BF dont la visibilité est évaluée par échantillonnage et par volumes de vision étendus.	107
5.3	Évaluation du nombre d'intersections calculées dans la scène de l'albisia en fonction du type d'intersections utilisé.	109
5.4	Évaluation du nombre d'intersections dans la scène du labyrinthe en fonction du type d'intersections utilisé.	110
5.5	Limitation des appels d'évaluation de la classification par volume de vision grâce à l'utilisation d'une sauvegarde du parcours du graphe de visibilité à un niveau précédent.	113
5.6	Tailles équivalentes d'obstacles sphériques.	116
5.7	Table récapitulative du nombre d'obstacles de chaque scène en fonction des paramètres de volume d'obstacles et de densité du cube global.	119

5.8	Résultats de l'outil de prédiction de la classification de visibilité basée sur une étude statistique sur la distribution des obstacles.	125
6.1	Temps de calcul de la reconstruction de la solution de radiativité en fonction de la méthode utilisée.	137

TABLE DES ALGORITHMES

3.1	Visualisation des avatars plans avec le stencil buffer d'Open GL	48
3.2	Regroupement en surfaces quasi coplanaires glouton	52
3.3	Regroupement en surfaces quasi coplanaires grâce à la hiérarchie	53
3.4	Regroupement en surfaces quasi coplanaires grâce à l'espace dual	56
3.5	Génération d'avatars à partir de surfaces quasi-coplanaires	59
5.1	Génération optimisée de volumes de vision	102
5.2	Création des plans à partir des arêtes	102
5.3	Calcul des numéros des sommets de l'arête déterminée par le couple (i, j)	104
5.4	Génération optimisée de volumes de vision étendus	105

BIBLIOGRAPHIE

- [AC97] Kendall Atkinson and Graeme Chandler. The collocation method for solving the radiosity equation for unoccluded surfaces. *Journal of Integral Equations & Applications*, 10(3) :253–260, 1997. Available from <http://www.math.uiowa.edu/atkinson/papers.html>.
- [AK89] James Arvo and David Kirk. A survey of ray tracing acceleration techniques. In Andrew S. Glassner, editor, *An introduction to ray tracing*, pages 201–262. Academic Press, 1989.
- [Arv93] J. Arvo. Transfert equations in global illumination. ACM SIGGRAPH course note. ACM Press N.Y., 1993. Appendix A.
- [Ash94] Ian Ashdown. *Radiosity : A Programmer's Perspective*. John Wiley & Sons, New York, NY, 1994.
- [Ash95] I. Ashdown. Photometry and radiometry. ACM SIGGRAPH course note. ACM Press N.Y., 1995. Appendix A.
- [Atk89] Kendall E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, second edition, 1989.
- [Bek99] Philippe Bekaert. *Hierarchical and Stochastic Algorithms for Radiosity*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 1999.
- [BG95] Christian-A. Bohn and Robert Garmann. A Parallel Approach to Hierarchical Radiosity. In V. Skala, editor, *Proceedings of the Winter School of Computer Graphics and CAD Systems '95*, pages 26–35, Plzen, Czech Republic, February 1995. University of West Bohemia.
- [BS96] Gonzalo Besuievsky and Mateu Sbert. The multi-frame lighting method : A monte carlo based solution for radiosity in dynamic environments. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 185–194, New York City, NY, June 1996. Eurographics, Springer Wien. ISBN 3-211-82883-4.
- [BW96] Philippe Bekaert and Yves Willems. Error control for radiosity. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 153–164, New York City, NY, June 1996. Eurographics, Springer Wien. ISBN 3-211-82883-4.

- [BWCG86] Daniel R. Baum, John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. The Back-Buffer Algorithm : An Extension of the Radiosity Method to Dynamic Environments. *The Visual Computer*, 2(5) :298–306, September 1986.
- [CAH00a] F. Cuny, L. Alonso, and N. Holzschuch. A novel approach makes higher order wavelets really efficient for radiosity. In M. Gross and F. R. A. Hopgood, editors, *Computer Graphics Forum (Eurographics 2000)*, volume 19(3), pages 99–108, 2000.
- [CAH00b] François Cuny, Laurent Alonso, and Nicolas Holzschuch. A novel approach makes higher order wavelets really efficient for radiosity. *Computer Graphics Forum*, 19(3) :C99–C108, Sep 2000. (Eurographics 2000 proceedings).
- [CAP98] Xavier Cavin, Laurent Alonso, and Jean-Claude Paul. Parallel wavelet radiosity. In *Proceedings of the Second Eurographics Workshop on Parallel Graphics and Visualisation*, pages 61–75, Rennes, France, September 1998. Eurographics.
- [Cat74] Edwin E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Ph.d. thesis, University of Utah, December 1974.
- [Cat75] Edwin E. Catmull. Computer display of curved surfaces. In *Proceedings of the IEEE Conference on Computer Graphics, Pattern Recognition, and Data Structure*, pages 11–17, May 1975.
- [Cav99] Xavier Cavin. Load balancing analysis of a parallel hierarchical algorithm on the origin2000. In *Fifth European SGI/Cray MPP Workshop*, Bologna, Italy, September 1999.
- [CCWG88] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. volume 22, pages 75–84, August 1988.
- [CDG⁺89] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, and H. Ralambondrainy. *Classification automatique des données*. Dunod Informatique, Paris, 1989.
- [Cel90] M. Celenk. A color clustering technique for image segmentation. *Computer Vision, Graphics, and Image Processing*, 52(2) :145–170, November 1990.
- [CF87] Sudeb Chattopadhyay and Akira Fujimoto. Bi-directional ray tracing. In Tsiyasu L. Kunii, editor, *Computer Graphics 1987 (Proceedings of CG International '87)*, pages 335–343. Springer-Verlag, 1987.
- [CG85] M. F. Cohen and D. P. Greenberg. The hemi-cube : A radiosity for complex environments. volume 19, pages 31–40, July 1985.
- [Che90] Shenchang Eric Chen. Incremental radiosity : An extension of progressive radiosity to an interactive image synthesis system. volume 24, pages 135–144, August 1990.
- [CLSS97] Per H. Christensen, Dani Lischinski, Eric J. Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1) :3–33, January 1997.
- [CMS87] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. *Computer Graphics (SIGGRAPH '87 Proc.)*, 21(4) :273–281, July 1987.
- [CR98] Alan Chalmers and Erik Reinhard. *SIGGRAPH 98 "Parallel and Distributed Photo-Realistic Rendering" course #3 notes*. July 1998.
- [Cun00] François Cuny. *Radiosité à base d'ondelettes sur des surfaces paramétriques*. Ph.d. thesis, Institut National Polytechnique de Lorraine, October 2000.

- [CW93] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, MA, 1993.
- [Dam01] Cyrille Damez. *Simulation globale de l'éclairage pour des séquences animées prenant en compte la cohérence temporelle*. PhD thesis, Université Joseph Fourier, Grenoble, France, 2001.
- [DDP97] Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton : A powerful and efficient multi-purpose global visibility tool. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 89–100. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [DDP99] Frédo Durand, George Drettakis, and Claude Puech. Fast and accurate hierarchical radiosity using global visibility. In *ACM Transactions on Graphics*, volume 18 (2), pages 128–170, 1999.
- [dl83] Commissions Internationale de l'Eclairage. The basis of physical photometry. Central Bureau of the CIE, Vienna, 1983. Publ. n° 18.2, 2nd edition.
- [DPC01] Yann Dupuy, Mathias Paulin, and Rene' Caubet. Occlusion evaluation in hierarchical radiosity. In V. Skala, editor, *WSCG 2001 Conference Proceedings*, 2001.
- [DS97] George Drettakis and François Sillion. Interactive update of global illumination using A line-space hierarchy. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 57–64. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [DS99] C. Damez and F. Sillion. Space-time hierarchical radiosity for high-quality animations. In *Rendering Techniques '99*, pages 235–246, New York, NY, 1999. Springer Wien.
- [Dur99] Fredo Durand. *3D Visibility : Analytical Study and Applications*. PhD thesis, Université Joseph Fourier, Grenoble, France, July 1999.
- [Dur00] Frédo Durand. A multidisciplinary survey of visibility. ACM SIGGRAPH course note. ACM Press N.Y., 2000.
- [DvGNK99] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *j-TOG*, 18(1) :1–34, January 1999.
- [DW94] Philip Dutre and Yves D. Willems. Importance-Driven Monte Carlo Light Tracing. In *Fifth Eurographics Workshop on Rendering*, pages 185–194, Darmstadt, Germany, June 1994.
- [Fed95] Martin Fedá. Speeding Up Progressive Radiosity by Overshooting. In V. Skala, editor, *Proceedings of the Winter School of Computer Graphics and CAD Systems '95*, pages 87–92, Plzen, Czech Republic, February 1995. University of West Bohemia.
- [FP92] Martin Fedá and Werner Purgathofer. Accelerating radiosity by overshooting. *Third Eurographics Workshop on Rendering*, pages 21–32, May 1992.
- [FvFH90] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer graphics : Principles and Practice*. Addison-Wesley, 2nd edition, 1990.
- [FYT94] David A. Forsyth, Chien Yang, and Kim Teo. Efficient Radiosity in Dynamic Environments. In *Fifth Eurographics Workshop on Rendering*, pages 313–323, Darmstadt, Germany, June 1994.
- [Geo90] David W. George. A Radiosity Redistribution Algorithm for Dynamic Environments. M.Sc. thesis, Ithaca, NY, August 1990.

- [Gla88] Andrew S. Glassner. Spacetime ray tracing for animation. *IEEE Computer Graphics and Applications*, 8(2) :60–70, March 1988.
- [Gla95] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.
- [GMN94] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman. Wavelength dependent reflectance functions. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 213–220. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [Gou71] H. Gouraud. Continuous shading of surfaces. *IEEE Transactions on Computers June 1971*, C-20(6) :223–228, 1971.
- [GSCH93] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 221–230, 1993.
- [GSG90] David W. George, François X. Sillion, and Donald P. Greenberg. Radiosity redistribution for dynamic environments. *IEEE Computer Graphics and Applications*, 10(4) :26–34, July 1990.
- [GSH94] Reid Gershbein, Peter Schroder, and Pat Hanrahan. Textures and Radiosity : Controlling Emission and Reflection with Texture Maps. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 51–58, 1994.
- [GTGB84] Cindy M. Goral, Kenneth K. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. volume 18, pages 213–222, July 1984.
- [Guo98] Baining Guo. Progressive radiance evaluation using directional coherence maps. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 255–266. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [HA00] Nicolas Holzschuch and Laurent Alonso. Using graphics hardware to speed-up visibility queries. *Journal of Graphics Tools*, 5(2) :33–47, 2000.
- [Hai00] Eric Haines. A shaft culling tool. *Journal of Graphics Tools : JGT*, 5(1) :23–26, 2000.
- [HCA00] Nicolas Holzschuch, Francois Cuny, and Laurent Alonso. Wavelet radiosity on arbitrary planar surfaces. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 161–172, New York, NY, 2000. Springer Wien.
- [HDS99] Jean-Marc Hasenfratz, Cyrille Damez, François Sillion, and George Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. In *Computer Graphics Forum (Proc. of Eurographics '99)*, number 18(3), pages 221–232. P. Brunet and R. Scopigno, Sep 1999.
- [Hec90] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. volume 24, pages 145–154, August 1990.
- [Hec91] P.S. Heckbert. Finite element methods for global illumination. Technical Report UCB/CSD 91/643, University of California, Berkeley, Computer Science Division (EECS), July 1991.

- [Hec94] Paul S. Heckbert, editor. *Graphics gems IV*, volume 4 of *Graphics Gems*. AP Professional, Boston, MA, USA, 1994.
- [Hol96] Nicholas Holzschuch. (*Le Controle de l'Erreur dans la Methode de Radiosite Hierarchique (Error Control in Hierarchical Radiosity)*). Ph.D. thesis, Grenoble, France, March 5th, 1996.
- [How82] John R. Howell. *A Catalog of Radiation Configuration Factors*. McGraw Hill, New York, NY, 1982.
- [HS67] Hoyt C. Hottel and Adel F. Sarofim. *Radiative Transfer*. McGraw Hill, New York, NY, 1967.
- [HS90] Pat Hanrahan and David Salzman. A rapid hierarchical radiosity algorithm for unoccluded environments. In *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 151–71, Rennes, France, June 1990.
- [HS98] Nicholas Holzschuch and Francois. X. Sillion. An exhaustive error-bounding algorithm for hierarchical radiosity. *Computer Graphics Forum*, 17(4) :197–218, December 1998.
- [HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4) :197–206, July 1991.
- [HSD94] Nicolas Holzschuch, Francois Sillion, and George Drettakis. An Efficient Progressive Refinement Strategy for Hierarchical Radiosity. In *Fifth Eurographics Workshop on Rendering*, pages 343–357, Darmstadt, Germany, June 1994.
- [Hut93] David J. Hutchinson. Monte Carlo Path Tracing for Photorealism. M.Sc. thesis, Manchester, UK, October 1993.
- [HW91] Eric Haines and John Wallace. Shaft culling for efficient ray-traced radiosity. In *Eurographics Workshop on Rendering*, 1991.
- [HW94] Eric A. Haines and John R. Wallace. Shaft Culling for Efficient Ray-Traced Radiosity. In P. Brunet and F. W. Jansen, editors, *Photorealistic Rendering in Computer Graphics (Proceedings of the Second Eurographics Workshop on Rendering)*, New York, NY, 1994. Springer-Verlag.
- [iOKS80] Yu ichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13(3) :222–241, July 1980.
- [JC95] Henrik Wann Jensen and Niels Jørgen Christensen. Photon maps in bidirectional Monte Carlo ray tracing of complex objects. *Computers and Graphics*, 19(2) :215–224, March–April 1995.
- [Jen95] Henrik Wann Jensen. Importance Driven Path Tracing Using the Photon Map. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 326–335, New York, NY, 1995. Springer-Verlag.
- [Kaj86] J. T. Kajiya. The rendering equation. volume 20, pages 143–150, August 1986.
- [Kel97] Alexander Keller. Instant radiosity. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 49–56. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [Kri47] E. L. Krinov. *Spectral reflectance properties of natural formations*. Laboratoria Aerometodov, Akad. Nauk SSSR, Moscow, 1947.

- [LAP00] Jean-Christophe Ulysse Laurent Alonso, Xavier Cavin and Jean-Claude Paul. Fast and accurate wavelet radiosity computations using high-end platforms. In *Proceedings of the Third Eurographics Workshop on Parallel Graphics and Visualisation*, pages 25–38, Universitat de Girona, Spain, September 2000.
- [Lov68] Tom J. Love. *Radiative Heat Transfer*. Charles E. Merrill Publishing, Columbus, OH, 1968.
- [LP00] Luc Leblanc and Pierre Poulin. Guaranteed occlusion and visibility in cluster hierarchical radiosity. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 89–100, New York, NY, 2000. Springer Wien.
- [LSG94] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. *Computer Graphics*, 28(Annual Conference Series) :67–74, July 1994.
- [LW93] Eric P. Lafortune and Yves D. Willems. Bi-directional Path Tracing. In H. P. Santo, editor, *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, December 1993.
- [MA92] Nelson L. Max and Michael J. Allison. Linear Radiosity Approximation Using Vertex-to-Vertex Form Factors. In David Kirk, editor, *Graphics Gems III*, pages 318–323. Academic Press Professional, Boston, MA, 1992.
- [Mal88] Thomas J. V. Malley. A Shading Method for Computer Generated Images. M.Sc. thesis, June 1988.
- [MBG86] Gregory M. Maxwell, Michael J. Bailey, and Victor W. Goldschmidt. Calculations of the Radiation Configuration Factor Using Ray Casting. *Computer-Aided Design*, 18(7) :371–379, September 1986.
- [MT97] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, 2(1), 1997. ISSN 1086-7651.
- [Neu94] Laszlo Neumann. New Efficient Algorithms with Positive Definite Radiosity Matrix. In *Fifth Eurographics Workshop on Rendering*, pages 219–237, Darmstadt, Germany, June 1994.
- [NTE95] Laszlo Neumann, Robert F. Tobler, and Pavol Elias. The Constant Radiosity Step. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 336–344, New York, NY, 1995. Springer-Verlag.
- [ORDP96] Rachel Orti, Stephane Riviere, Fredo Durand, and Claude Puech. Radiosity for Dynamic Scenes in Flatland with the Visibility Complex. In *Computer Graphics Forum*, volume 15, pages C237–C248, September 1996.
- [Pho75] Bui-Tuong Phong. Illumination for computer generated pictures. *CACM June 1975*, 18(6) :311–317, 1975.
- [PP98] Ingmar Peter and Georg Pietrek. Importance driven construction of photon maps. In George Drettakis and Nelson Max, editors, *Rendering Techniques '98*, Eurographics, pages 269–280. Springer-Verlag Wien New York, 1998.
- [RAPP97] Luc Renambot, Bruno Arnaldi, Thierry Priol, and Xavier Pueyo. Towards efficient parallel radiosity for DSM-based parallel computers using virtual interfaces. In James Painter, Gordon Stoll, and Kwan-Liu Ma, editors, *IEEE Parallel Rendering Symposium*, pages 79–86. IEEE, November 1997. ISBN 1-58113-010-4.

- [RCJ98] Erik Reinhard, Alan Chalmers, and Frederik W. Jansen. Overview of parallel photo-realistic graphics. Technical Report CS-EXT-1998-147, University of Bristol, UK. Computer Science, August 1, 1998.
- [RGG90] Rodney J. Recker, David W. George, and Donald P. Greenberg. Acceleration techniques for progressive refinement radiosity. In *Computer Graphics (Proceedings of the 1990 Symposium on Interactive 3D Graphics)*, pages 59–66, March 1990. Cornell University.
- [Ron56] Vasco Ronchi. *Histoire de la lumière*. Jacques Gabay Edition, 1956.
- [SAG94] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [SAS92] Brian E. Smits, James R. Arvo, and David H. Salesin. An importance-driven radiosity algorithm. *Computer Graphics*, 26(2) :273–282, July 1992.
- [SAWG91] Francois X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. volume 25, pages 187–196, July 1991.
- [Sbe97] Mateu Sbert. *The Use of Global Random Directions to Compute Radiosity : Global Monte Carlo Techniques*. PhD thesis, Universitat Politecnica de Catalunya, Barcelona, Spain, 1997. Available from <http://ima.udg.es/mateu>.
- [SGCH93] Peter Schroder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. Wavelet Projections for Radiosity. In *Fourth Eurographics Workshop on Rendering*, number Series EG 93 RW, pages 105–114, Paris, France, June 1993.
- [SGL94] Jaswinder Pal Singh, Anoop Gupta, and Marc Levoy. Parallel visualization algorithms : Performance and architectural implications. *Computer*, 27(7) :45–55, July 1994.
- [SH00] Francois Sillion and J.-M. Hasenfratz. Efficient parallel refinement for hierarchical radiosity on a DSM computer. In *Proceedings of the Third Eurographics Workshop on Parallel Graphics and Visualisation*, pages 61–74, Universitat de Girona, Spain, September 2000. Available from <http://www-imagis.imag.fr/Jean-Marc.Hasenfratz/PUBLI/EGWPGV00.html>.
- [Sha97] Erin Shaw. Hierarchical radiosity for dynamic environments. *Computer Graphics Forum*, 16(2) :107–118, 1997. ISSN 0167-7055.
- [Sil94] Francois Sillion. Clustering and Volume Scattering for Hierarchical Radiosity Calculations. In *Fifth Eurographics Workshop on Rendering*, pages 105–117, Darmstadt, Germany, June 1994.
- [Sil95] Francois X. Sillion. A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3) :240–254, September 1995.
- [Sol98] Cyril Soler. *Hierarchical Representations of Visibility for Error Control in Lighting Simulations*. PhD thesis, Universite de Joseph Fourier, Grenoble, France, December 1998.
- [SP89] Francois Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. volume 23, pages 335–344, July 1989.

- [SP94] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, 1994. excellent coverage of radiosity and global illumination algorithms.
- [SP95] Mateu Sbert and Xavier Pueyo. Integral Geometry Methods for Form Factor Computation. *Encuentros de Geometria Computacional*, July 1995.
- [SS95] Peter Schröder and Wim Sweldens. Spherical wavelets : Efficiently representing functions on the sphere. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 161–172. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.
- [SS96] Cyril Soler and François Sillion. Accurate error bounds for multi-resolution visibility. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 133–142, New York City, NY, June 1996. Eurographics, Springer Wien. ISBN 3-211-82883-4.
- [SSG⁺99] Marc Stamminger, Annette Scheel, Xavier Granier, Frederic Perez-Cazorla, George Drettakis, and François Sillion. Efficient glossy global illumination with interactive viewing. In I. Scott MacKenzie and James Stewart, editors, *Proceedings of the Conference on Graphics Interface (GI-99)*, pages 50–57, Toronto, Ontario, June 2–4 1999. CIPS.
- [SSPS01] Annette Scheel, Marc Stamminger, Jörg Pütz, and Hans-Peter Seidel. Enhancements to directional coherence maps. In *Ninth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG 2001)*, Plzen, Czech Republic, February 2001. University of West Bohemia. Available from <http://wscg.zcu.cz/wscg2001>.
- [SSS01] Annette Scheel, Marc Stamminger, and Hans-Peter Seidel. *Thrifty Final Gather Radiosity*. Springer Wien, New York, NY, 2001. To appear.
- [SSSS98] M. Stamminger, H. Schirmacher, P. Slusallek, and Hans-Peter Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Forum*, 17(3) :165–174, 1998. ISSN 1067-7055.
- [SW99] F. Suykens and Y. D. Willems. Combining bidirectional path tracing and multipass rendering. In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)*, pages 265–272, Plzen-Borey, Czech Republic, February 1999. University of West Bohemia.
- [The97] J. Theiler. A contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation, 1997.
- [Tou83] G. Toussaint. Solving geometric problems with the rotating calipers. volume 10, pages 1–4, May 1983.
- [VG94] Eric Veach and Leonidas Guibas. Bidirectional Estimators for Light Transport. In *Fifth Eurographics Workshop on Rendering*, pages 147–162, Darmstadt, Germany, June 1994.
- [War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. volume 26, pages 265–272, July 1992.
- [WAT92] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. volume 26, pages 255–264, July 1992.

- [WCG87] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation : A synthesis of ray tracing and radiosity methods. *Computer Graphics*, 21(4) :311–320, July 1987.
- [WH97] Andrew J. Willmott and Paul S. Heckbert. An empirical comparison of progressive and wavelet radiosity. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 175–186, New York City, NY, June 1997. Eurographics, Springer Wien. ISBN 3-211-83001-4.
- [WHG99] Andrew J. Willmott, Paul S. Heckbert, and Michael Garland. Face cluster radiosity. In Dani Lischinski and Greg Ward Larson, editors, *Rendering Techniques '99*, Eurographics, pages 293–304. Springer-Verlag Wien New York, 1999.
- [WS82] G. Wyszecki and W. S. Stiles. *Color Science : Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, 1982. second edition.
- [WW92] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques*. Addison-Wesley, 1992. ISBN 0–201–54412–1.
- [Zat93] Harold R. Zatz. Galerkin radiosity : A higher order solution method for global illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 213–220, 1993.