



HAL
open science

Simulation globale de l'éclairage pour des séquences animées prenant en en compte la cohérence temporelle

Cyrille Damez

► **To cite this version:**

Cyrille Damez. Simulation globale de l'éclairage pour des séquences animées prenant en en compte la cohérence temporelle. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2001. Français. NNT: . tel-00004671

HAL Id: tel-00004671

<https://theses.hal.science/tel-00004671v1>

Submitted on 16 Feb 2004

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.

Simulation globale de l'éclairage pour des séquences animées prenant en compte la cohérence temporelle

Cyrille DAMEZ

Thèse présentée pour l'obtention du titre de Docteur de l'Université Joseph Fourier
Spécialité Informatique

Arrêté ministériel du 5 juillet 1984 et du 30 mars 1992

Préparée au sein du laboratoire *i*MAGIS-GRAVIR/IMAG-INRIA. UMR CNRS C5527.

Soutenue publiquement le 10 Décembre 2001

Composition du jury :

Didier	ARQUÈS	Président
Pierre	POULIN	Rapporteur
Xavier	PUEYO	Rapporteur
Claude	PUECH	Examineur
François	SILLION	Directeur de thèse

Remerciements

Je voudrais tout d'abord remercier Claude Puech pour m'avoir accueilli dans son équipe. De l'avis de tous, *iMAGIS* lui est très largement redevable pour l'excellent niveau des recherches qui y sont menées et la très bonne ambiance de travail qui y règne.

Je souhaite également exprimer ma reconnaissance à Pierre Poulin et Xavier Pueyo pour avoir accepté d'être les rapporteurs de cette thèse. Leurs remarques pertinentes sur son contenu me seront précieuses pour la suite de mes recherches. Ma gratitude va également à Didier Arquès pour avoir accepté de présider ce jury, malgré les contraintes liées à un emploi du temps très chargé.

François Sillion a été pour moi le directeur de thèse que beaucoup souhaiteraient avoir. Sa très grande connaissance du domaine, ses capacités d'analyse rapide et de synthèse m'ont été très précieuses. J'ai eu énormément de plaisir à travailler avec lui.

L'équipe *iMAGIS* est particulièrement accueillante, et ses succès dans la recherche sont à la mesure du plaisir que l'on a à y travailler. J'ai au cours de ces trois années, pu bénéficier des conseils avisés et de la collaboration de ses membres et de ses invités. Je tiens particulièrement à remercier Cyril et Nicolas pour les relectures d'un document dont l'accouchement fut laborieux, Alex avec qui j'ai partagé les affres de l'apprenti-docteur, Xavier pour son aide, scientifique et extra-scientifique, tout au long de cette thèse, ainsi qu'Eric et Jérémie avec qui je me suis souvent réfugié aux Caraïbes dans le monde de la flibuste, lorsque le temps était morose.

Il n'a pas du être facile de supporter un thésard comme colocataire. Les horaires aléatoires m'ont détourné plus que de raison des Travaux d'Intérêt Commun ; le moral en dent de scie et l'obsession pour ma discipline, si fréquents chez le doctorant, ont plombé les conversations au cours de plus d'un repas. Que soient ici remerciés pour leur patience et leur soutien, dans l'ordre chronologique, Marco, Eva, Ingo, Gaëlle, Ella, Petko, Mirella, Christina, Laurent, Javier, Pauline, Frank et Diego (ouf...).

Pour avoir trop traîné au *Bar*, en compagnie de musiciens de toutes nationalités, j'ai développé un certain goût pour le hors-sujet. Qu'ils soient ici remerciés pour avoir su me sortir périodiquement de l'ennuyeuse monomanie dont souffrent tous les thésards. Je salue également mes compagnons du *Réfrigérateur*, et tout particulièrement Fred, dont les oreilles peu chastes ont supporté mes borborygmes électroniques au mépris du bon goût et au péril de leurs tympans. Un grand merci également aux camarades du 102 pour nous avoir ouvert leur cave, mais aussi les yeux et les oreilles.

Finalement, je voudrais remercier mes parents pour leur soutien, mon frère qui a décidé d'entrer dans la carrière alors que son aîné n'y est plus (espérons qu'il n'y trouvera pas trop de poussière) et Régis, dont l'amitié ne craint ni le temps ni la distance.

Introduction

UN des premiers objectifs formulés de la recherche en synthèse d'image a été le *photoréalisme*. Les progrès conjoints du matériel et des algorithmes développés ont en effet rapidement permis d'espérer obtenir des images d'un réalisme tel qu'on ne saurait les distinguer d'une photographie.

Bien évidemment, les efforts fournis en ce sens ne l'ont pas été par simple goût de la difficulté. De nombreuses applications industrielles de l'imagerie de synthèse existent, qui requièrent une reproduction aussi fidèle que possible de la réalité. En particulier, l'industrie du cinéma, et celle bientôt plus importante encore des jeux vidéos, s'appuient de plus en plus sur les techniques de production d'images par ordinateur. Qu'il s'agisse de l'inclusion d'acteurs ou de décors virtuels, ou bien encore de la réalisation d'effets spéciaux, l'œil du spectateur ne doit pas pouvoir discerner la réalité de sa contrefaçon. De plus, si les premiers films intégralement composés d'image de synthèse s'inspiraient de l'esthétique des dessins animés, les productions les plus récentes se sont montrées nettement plus ambitieuses quant au degré de réalisme recherché.

Dans la quête du photoréalisme, la piste la plus prometteuse est celle de la simulation de l'éclairage. Puisque une image, qu'elle se forme sur la rétine de l'œil ou sur pellicule, est essentiellement une mesure du flux lumineux à travers une surface, la modélisation et la simulation numérique de la circulation de la lumière devrait nous permettre d'approcher au plus près la réalité. Pour cela, différents problèmes doivent être résolus.

Dans un premier temps, il est nécessaire de construire un modèle mathématique qui permette de traduire le comportement de la lumière en équations. Pour ce faire, nous disposons des résultats de siècles de recherche en optique physique et géométrique. Encore faut-il choisir un modèle approprié aux simulations que l'on souhaite conduire. Pour notre part, nous utiliserons un modèle se basant essentiellement sur les lois de l'optique géométrique. Les phénomènes dont ce modèle ne peut rendre compte (interférence, diffraction) n'interviennent en effet que rarement dans la synthèse d'une image.

Une fois choisi le modèle physique, il faut rassembler les données, éventuellement issues du monde réel, nécessaires à la poursuite de la simulation. La question de l'acquisition des données est un problème à part entière, qu'il s'agisse de la géométrie des objets (en utilisant un modéleur, un scanner 3D, *etc.*), des propriétés physiques des matériaux (réflexion, transparence, reliefs de la surface, *etc.*), des conditions d'éclairage artificielles (modélisation des lampes) ou naturelles (modélisation du ciel, acquisition des conditions d'éclairage pour l'incrustation dans une image réelle). De plus,

lorsque l'on souhaite effectuer le calcul d'une animation, il faut également déterminer le mouvement des objets dans l'espace. Cela peut se faire soit en spécifiant « manuellement » les trajectoires des points, ou après acquisition sur un modèle réel (capture de mouvement), soit par simulation des lois de la mécanique, les données étant alors les forces en présence, ou leurs causes.

Dans ce mémoire, nous délaierons, à regret, les questions de première importance soulevées par l'acquisition des données pour nous concentrer sur le problème de la simulation numérique des flux lumineux. L'exigence de réalisme nous oblige à prendre en compte le fait que la lumière que nous percevons n'est pas toujours issue directement de sources de lumière intrinsèques (le soleil, une lampe, *etc.*) mais parvient à notre œil après de multiples réflexions sur les surfaces environnantes. La détermination complète des échanges lumineux entre objets est depuis une quinzaine d'années un des problèmes centraux de la synthèse d'image désigné sous le nom de « simulation de l'éclairage global ». C'est à certains aspects de cette problématique que se consacre ce mémoire.

Plus spécifiquement, nous nous intéresserons au développement d'une méthode effectuant cette simulation dans le cadre d'une animation. En effet, si de telles simulations ont été pendant longtemps uniquement destinées à la seule production d'images fixes, la puissance de calcul actuellement à la disposition des studios de production¹ et le perfectionnement des algorithmes ont atteint un niveau qui rend envisageable leur utilisation pour la production de séquences animées.

Nous présenterons au chapitre 1 le modèle physique sur lequel nous nous appuierons. Nous y rappellerons les méthodes les plus couramment employées en synthèse d'image pour la simulation de l'éclairage global. Nous développerons plus particulièrement le principe des algorithmes dits « de radiosité », sur lequel se base l'algorithme de radiosité hiérarchique spatiale et temporelle, principale contribution de cette thèse. Nous y verrons que, si les résultats produits par ces algorithmes ne peuvent pas toujours être utilisés directement pour la production d'images esthétiquement satisfaisantes, ils peuvent en revanche être utilisés pour fournir l'éclairage ambiant à un algorithme de rendu local lors d'une seconde passe. Les images ainsi obtenues sont généralement d'un réalisme surprenant.

Nous nous intéresserons dans le chapitre suivant aux spécificités du problème de l'éclairage global dans le cadre du calcul d'animations. Le problème posé est d'une double nature : il ne s'agit pas uniquement de produire une animation de très haute qualité, mais également, comme pour tout procédé de production, de permettre de le faire pour un coût le plus faible possible. Nous montrerons dans ce chapitre que ces deux problématiques nous amènent à nous intéresser à la continuité temporelle des échanges lumineux. D'une part, nous montrerons que cette dernière est suffisamment importante pour nous permettre de réduire la quantité de calculs nécessaires à l'éclairage d'une scène animée. D'autre part, nous verrons que la continuité temporelle même de la solution est un des points essentiels dans la qualité perçue de l'animation.

Nous passerons également en revue dans ce chapitre les principales méthodes existant à ce jour permettant le calcul de l'éclairage global pour une scène animée. En conclusion de ce bref état de l'art, il apparaît qu'une méthode hiérarchique basée sur l'algorithme de radiosité, destinée à la simulation de l'éclairage global, pour la production d'une animation de qualité reste encore à réaliser. L'algorithme de radiosité hiérarchique, que nous détaillerons dans les deux parties suivantes, vise à produire une telle simulation.

La plupart des méthodes d'éclairage appliquées aux animations considèrent le problème posé comme une succession d'équations séparées à résoudre, correspondant à chacune des images de l'animation. Afin de mettre à profit la cohérence temporelle, et la connaissance des trajectoires des objets à notre disposition lors du calcul d'un film de synthèse, nous raisonnerons directement dans un espace à

¹et, dans une moindre mesure, des particuliers

quatre dimensions, produit cartésien de l'espace géométrique classique et de la direction temporelle. Ainsi, il nous sera plus aisé de raisonner sur la notion de continuité temporelle de la solution que si nous avons modélisé la dimension temporelle de façon discrète.

Nous reformulerons dans cet espace, au chapitre 3, l'équation de radiosité. Nous ramènerons ainsi le problème de l'éclairage diffus dans une scène animée à la résolution d'une unique équation intégrale de nature similaire à l'équation statique de la radiosité. De ce fait, la plupart des méthodes employées pour le calcul de la radiosité pour une seule image pourront être directement transposées au cas des animations. Nous montrerons au chapitre 4 comment adapter l'algorithme classique de radiosité hiérarchique au cas des animations. L'élément le plus délicat pour la mise au point de cet algorithme est la définition d'un oracle de raffinement, nécessaire à la construction progressive du maillage et de la hiérarchie de liens.

Les premiers résultats, encourageants, obtenus par cette méthode seront présentés au chapitre 5. Nous y mettrons également en évidence ses limitations, que nous adresserons dans la troisième partie de ce mémoire. Les contributions présentées dans cette seconde partie ont donné lieu à une publication en 1999 au « *Workshop* » Eurographics sur le Rendu [26].

Dans les chapitres 6 et 7, nous verrons comment, en permettant la construction d'une hiérarchie de groupes au-dessus des surfaces, il est possible d'employer l'algorithme de radiosité hiérarchique spatiale et temporelle pour des scènes d'une complexité beaucoup plus importante. Il s'agit d'une version étendue de l'article abrégé [27] présenté à la session *Short Papers* de l'édition 2001 de la conférence Eurographics.

Nous montrerons, au chapitre suivant, que l'utilisation d'une base de fonction d'ordre supérieur nous permet d'améliorer la continuité, et donc la qualité, des animations obtenues. Finalement, nous présenterons au dernier chapitre une réorganisation inédite de la traversée de la hiérarchie permettant de mettre à contribution la mémoire de masse pour réduire la consommation en mémoire vive de notre algorithme.

Première partie

Travaux antérieurs et motivation

Illumination globale par la méthode de radiosit 

DANS ce chapitre, nous offrons une rapide introduction aux m thodes de synth se d'images par la simulation de l' clairage global. Contrairement aux premi res m thodes d velopp es dans le domaine qui consid rent l' clairage de chaque surface comme autant de probl mes ind pendants [33], les m thodes d' clairage global permettent, au prix d'une augmentation notable de la complexit  des algorithmes de r solution, de prendre en consid ration les effets dus aux multiples r flexions de la lumi re.

Dans un premier temps, nous pr senterons rapidement les mod les physiques et  quations couramment employ s en synth se d'images pour d crire les  changes lumineux entre objets. Nous insisterons particuli rement sur les hypoth ses de simplification effectu es et sur leurs cons quences sur le pouvoir descriptif des dits mod les. Nous ferons ensuite une pr sentation d taill e de la m thode de *radiosit * et de diff rentes am liorations apport es   l'algorithme original et plus particuli rement la m thode de *radiosit  hi rarchique*.

1.1 Introduction   la simulation de l' clairage

Les images telles que nous les percevons sont dues   la r ception d'un flux lumineux sur la r tine de l' cil. Il para t donc naturel, afin de g n rer des images aussi r alistes que possible, de chercher   simuler ce flux lumineux   l'aide d'un mod le physique ad quat. La pr sentation par Kajiya de l' quation du rendu [66] a permis d'unifier dans un m me formalisme (issu du calcul int gral) la plupart des algorithmes propos s jusqu'alors pour la simulation de l' clairage en synth se d'images. Nous pr sentons dans ce paragraphe un mod le math matique permettant de d crire les  changes lumineux. Ce mod le repose sur les lois de l'optique g om trique. Il ne peut donc rendre correctement compte des ph nom nes d    la nature ondulatoire des radiations lumineuses, tels que la diffraction,

les interférences ou la polarisation (ce dernier point n'est pas réellement pénalisant puisque l'œil humain n'est généralement pas sensible à la polarisation de la lumière).

Dans un premier temps, nous précisons les notations et définitions que nous utiliserons dans ce mémoire. Nous donnerons ensuite une expression générale de l'équation du rendu. Après avoir précisé les conséquences des hypothèses de simplification utilisées, nous reformulerons cette équation sous une forme plus simple. Nous terminerons par une analyse des propriétés mathématiques de cette équation à l'aide des outils offerts par l'analyse fonctionnelle [7].

1.1.1 Définitions et notations

Description de la géométrie

– *Points et surfaces :*

Un point de l'espace sera généralement repéré par ses coordonnées à valeur dans \mathbb{R}^3 par rapport à un repère fixe. Toutefois on souhaitera parfois repérer un point particulier par rapport à sa position sur une surface ou dans un volume paramétré. On utilisera alors un vecteur de la forme $(u, v) \in [0, 1]^2$ ou $(u, v, w) \in [0, 1]^3$ pour le situer sur cette surface ou dans ce volume. On appellera *scène* l'ensemble \mathcal{M} des paramètres des points situés soit dans un volume participatif, soit à la surface frontière d'un objet. On supposera disposer d'une fonction de conversion $M : \mathcal{M} \rightarrow \mathbb{R}^3$ permettant d'obtenir la position de n'importe quel point de la scène dans l'espace à partir de ses paramètres. Pour des raisons de lisibilité, nous assimilerons fréquemment \mathcal{M} à son image $M(\mathcal{M}) \subset \mathbb{R}^3$.

– *Angles solides :*

La notion d'angle solide est une extension de l'angle en dimension 3 : l'angle solide sous lequel une surface S est vue d'un point P est l'aire ω de la projection de S sur la sphère unité centrée en P (cf. figure 1.1).

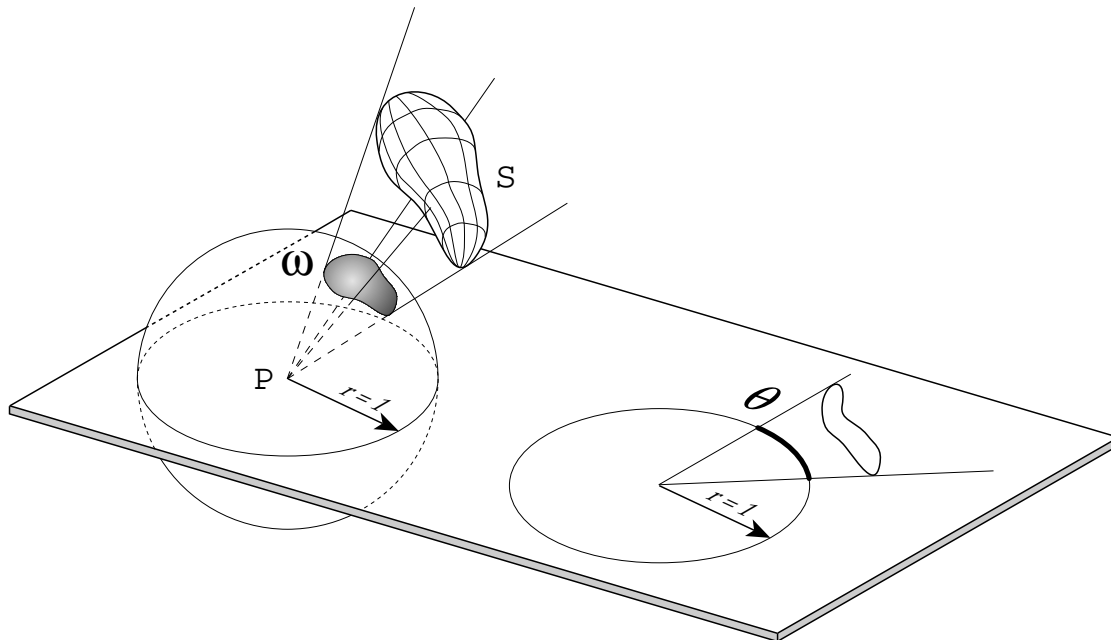


FIG. 1.1: Définition de l'angle solide et illustration de l'analogie avec la notion d'angle en dimension 2¹

¹Cette figure est extraite du livre de François Sillion et Claude Puech *Radiosity and Global Illumination* [104]

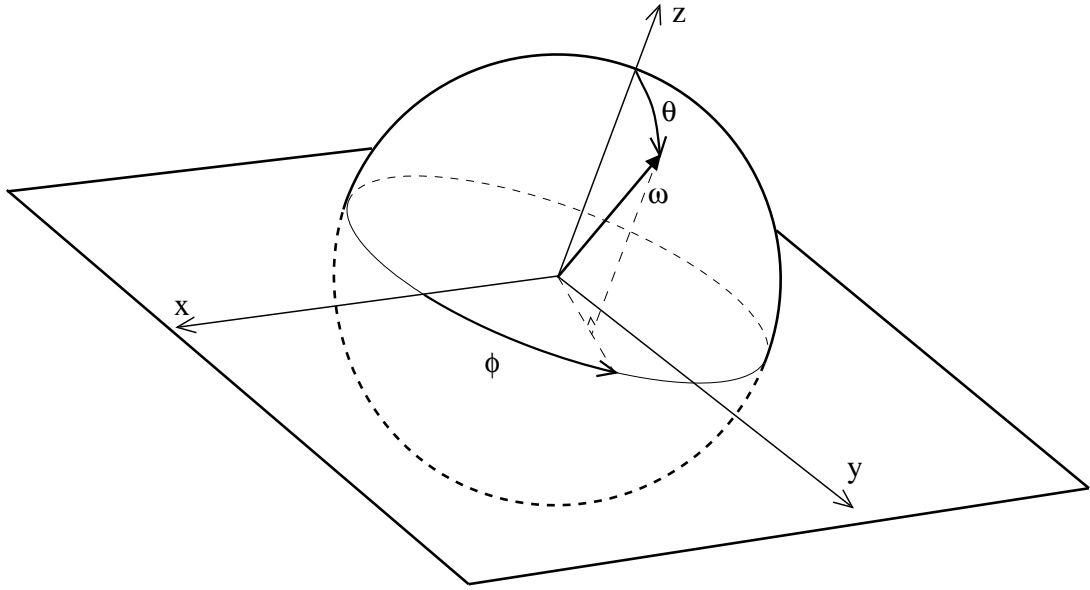


FIG. 1.2: Direction en coordonnées polaires

– *Directions :*

Nous représenterons dans cette thèse les directions comme des vecteurs normalisés appartenant à la sphère unité $S \subset \mathbb{R}^3$. Dans le cas où elles seront définies par référence à un repère local attaché à une surface, nous emploierons de préférence leurs coordonnées polaires $(\theta, \phi) \in [0, \pi] \times [0, 2\pi[$, comme indiqué par la figure 1.2. On notera également S^+ la demi-sphère unité correspondant aux directions $(\theta, \phi) \in [0, \frac{\pi}{2}] \times [0, 2\pi[$.

Nous noterons parfois, afin d'éviter toute ambiguïté, le sens de propagation considéré : si la surface est considérée comme un récepteur, on notera la direction ω^\downarrow , comme un émetteur ω^\uparrow .

Quantification de la lumière

La lumière est une onde électromagnétique perçue dans une certaine plage de fréquences par l'œil humain. La variété des fréquences lumineuses se traduit alors par la notion de couleur. Lorsque les grandeurs dont nous rappellerons la définition ici seront définies pour une longueur d'onde précise λ , elle sera notée en exposant.

On appelle *luminance spectrale* l'énergie émise dans une certaine direction ω , pour une longueur d'onde λ par unité de temps, par unité d'angle solide et par unité de surface perpendiculaire à la direction de propagation. On la notera $L^\lambda(x, \omega)$. Ainsi, la puissance lumineuse irradiée à travers une surface différentielle d'aire d^2x dans la direction $\omega = (\theta, \phi)$ à travers l'angle solide $d\omega$, pour une longueur d'onde λ , correspondra à :

$$d^2P^\lambda(x, \omega) = L^\lambda(x, \omega) \cos \theta \, dx \, d\omega$$

On appelle *luminance* la somme des luminances spectrales sur l'ensemble du spectre lumineux.

$$L(x, \omega) = \int_0^\infty L^\lambda(x, \omega) \, d\lambda$$

On notera que, en l'absence de milieux participatifs, la luminance est invariante le long d'un rayon lumineux et en particulier ne s'atténue pas avec la distance. La connaissance de cette fonction pour

tous les points de la scène, dans la direction du récepteur (œil ou caméra), est donc suffisante pour la production d'une image photoréaliste [104]. Ainsi, en l'absence de milieu participatif, il est possible de produire des images d'un objet réel à partir de différents points de vue en échantillonnant le flux lumineux dans l'espace des rayons (on parle alors de fonction *plénoptique*) [1, 74, 45, 68].

La puissance lumineuse totale irradiée par une surface différentielle dx est :

$$dP(x) = dx \int_S L(x, \omega^\uparrow) \cos \theta^\uparrow d\omega^\uparrow$$

On appelle *radiosité* la puissance lumineuse totale $B(x)$ quittant un point par unité de surface :

$$\begin{aligned} B(x) &= \frac{dP}{dx} \\ &= \int_0^\infty \int_S L^\lambda(x, \omega^\uparrow) \cos \theta^\uparrow d\omega^\uparrow d\lambda \end{aligned} \quad (1.1)$$

De même, on appelle *irradiance* la puissance lumineuse totale $I(x)$ reçue en un point par unité de surface :

$$I(x) = \int_0^\infty \int_S L^\lambda(x, \omega^\downarrow) \cos \theta^\downarrow d\omega^\downarrow d\lambda \quad (1.2)$$

Redistribution de la lumière

Si l'on suppose que, pour une longueur d'onde donnée l'énergie lumineuse émise en un point est une fonction linéaire de l'énergie reçue en ce point dans toutes les directions de \mathcal{S} , on peut modéliser de façon générale les interactions de l'énergie lumineuse avec le milieu en employant la notion de *fonction de répartition bidirectionnelle* (Bidirectional Scattering Distribution Function) [58]. Cette dernière permet de décrire aussi bien les phénomènes de réflexion (*Bidirectional Reflectance Distribution Function* [79]) que de transmission (*Bidirectional Transmission Distribution Function*).

La *BSDF* $\rho_{bds}^\lambda(x, \omega^\downarrow, \omega^\uparrow)$ est définie comme le quotient du flux lumineux incident en x dans la direction ω^\downarrow avec le flux lumineux sortant au même point dans la direction ω^\uparrow :

$$\rho_{bds}^\lambda(x, \omega^\downarrow, \omega^\uparrow) = \frac{L^\lambda(x, \omega^\downarrow)}{L^\lambda(x, \omega^\uparrow) \cos \theta^\uparrow d\omega^\uparrow} \quad (1.3)$$

Elle s'exprime donc par unité d'angle solide (sr^{-1}).

On distinguera trois cas particuliers :

- La transmission idéale (par exemple dans le vide)
- Les réflecteurs *purement spéculaires*, qui réfléchissent la lumière dans la direction du miroir indiquée par la Loi de Descartes :

$$\rho_{bds}^\lambda(x, \omega^\downarrow, \omega^\uparrow) = \rho_s^\lambda(x, \theta^\uparrow) 2\delta(\sin^2 \theta^\downarrow - \sin^2 \theta^\uparrow) \delta(\phi^\downarrow - \phi^\uparrow \pm \pi) \quad (1.4)$$

- Les réflecteurs *purement diffus* : dans ce cas la *BSDF* est une fonction indépendante des termes angulaires ω^\downarrow et ω^\uparrow , et donc l'aspect de la surface de réception ne change pas avec l'angle de vue.

$$\forall (\omega^\downarrow, \omega^\uparrow) \in \mathcal{S}^+, \rho_{bds}^\lambda(x, \omega^\downarrow, \omega^\uparrow) = \frac{\rho^\lambda(x)}{\pi} \quad (1.5)$$

Récapitulatif

Le tableau 1.1 récapitule les symboles et unités des grandeurs définies dans les paragraphes précédents.

Grandeur	Symbole	Unité S.I.
Luminance	L	$W.m^{-2}.sr^{-1}$
Radiosité	B	$W.m^{-2}$
Irradiance	I	$W.m^{-2}$
BSDF	ρ_{bds}	sr^{-1}
Réfectance	ρ	.

TAB. 1.1: Récapitulatif des grandeurs et unités

1.1.2 L'équation du rendu

Nous allons dans cette partie donner une expression générale de la luminance dans une scène, obtenue à partir de la formulation de l'équilibre énergétique dans les échanges lumineux. Puis nous simplifierons cette expression en formulant une série d'hypothèses raisonnables (dont l'hypothèse classique dite « *de radiosité* »). Finalement, nous procéderons à une analyse formelle des équations présentées afin de discuter l'existence de solutions.

Équilibre énergétique

La luminance en un point x de l'espace, émise dans la direction ω^\uparrow est, à l'équilibre, la somme de la luminance créée en ce point et de la luminance totale réémise dans cette direction, ce que résume l'équation suivante :

$$L^\lambda(x, \omega^\uparrow) = L_{créée}^\lambda(x, \omega^\uparrow) + L_{redistribuée}^\lambda(x, \omega^\uparrow) \quad (1.6)$$

La luminance peut être créée par différents mécanismes (incandescence, luminescence, *etc.*) dont l'étude sort du cadre de ce document. Nous supposons simplement connue la fonction $L_{créée}^\lambda$. Le second terme de cette somme peut être exprimé comme la somme des luminances provenant de toutes les directions redistribuées dans la direction ω^\uparrow conformément à la définition de la *BSDF* (cf. équation 1.3) :

$$L_{redistribuée}^\lambda(x, \omega^\uparrow) = \int_S \rho_{bds}^\lambda(x, \omega^\downarrow, \omega^\uparrow) L^\lambda(x, \omega^\downarrow) \cos \theta^\downarrow d\omega^\downarrow$$

Or nous avons vu au paragraphe 1.1.1 que la luminance est invariante le long d'un rayon lumineux dans le vide. On peut donc exprimer la luminosité incidente $L^\lambda(x, \omega^\downarrow)$ en faisant référence à un autre point de la scène :

$$\forall z \in [x, y] L^\lambda(x, \omega^\downarrow) = L^\lambda(z, \omega^\downarrow)$$

où le segment $[x, y]$ est le segment maximal de la droite passant par x dans la direction ω^\downarrow entièrement hors milieu participatif. L'obtention du point y correspond à l'opération classique en synthèse d'images de *tracé de rayon* et peut être défini formellement [7] en employant une fonction $d : (\mathbb{R}^3 \times \mathcal{S}) \rightarrow \mathbb{R}^+$ qui indique la distance maximum pouvant être parcourue dans le vide à partir de x , ou, en d'autres termes, la distance du point le plus proche de x appartenant à la scène :

$$d(x, \omega^\downarrow) = \inf_l \{l \in \mathbb{R}^+; (x - l.\omega^\downarrow) \in \mathcal{M}\}$$

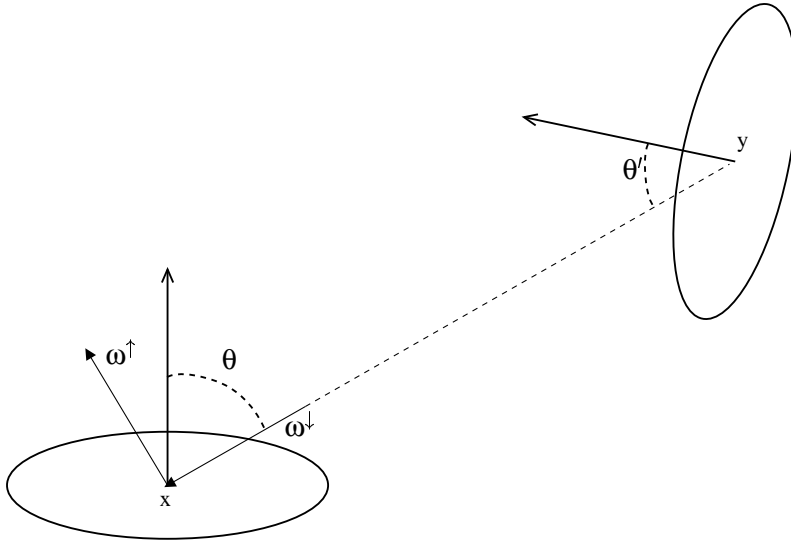


FIG. 1.3: Notations pour l'équation 1.7

On peut alors définir la fonction de *visibilité*

$$v : \begin{cases} (\mathbb{R}^3 \times \mathbb{R}^3) & \rightarrow \{0, 1\} \\ v(x, y) & = 1 \text{ si } y = x - \alpha \cdot \omega^\downarrow \text{ et } \alpha \leq d(x, \omega^\downarrow) \\ & = 0 \text{ sinon} \end{cases}$$

L'équation 1.6 se réécrit alors sous la forme d'une équation *globale*, c'est-à-dire faisant référence pour définir la luminance en chaque point, à tous les autres points de la scène. Il faut pour cela, dans l'intégrale, effectuer le changement de variable de l'angle solide $d\omega^\downarrow$ en élément de surface dy . Cela amène :

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

avec θ' défini par la figure 1.3.

$$L^\lambda(x, \omega^\uparrow) = L_{créée}^\lambda(x, \omega^\uparrow) + \int_{\mathcal{M}} \rho_{bds}^\lambda(x, \omega^\downarrow, \omega^\uparrow) L^\lambda(y, \omega^\downarrow) v(x, y) \frac{\cos \theta \cos \theta'}{\|x - y\|^2} dy \quad (1.7)$$

Hypothèses simplificatrices : De l'expression de la luminance à l'équation du rendu diffuse

Il n'existe malheureusement pas de solution analytique à l'équation 1.7 dans le cas général. Il faudra donc s'en remettre à des mécanismes de résolution numériques. Comme nous le verrons dans la section suivante, cette équation est d'une forme relativement classique et de nombreuses méthodes adaptées ont été mises au point à cet effet, notamment dans le domaine de l'ingénierie thermique. Bien que l'utilisation de cette équation sous sa forme la plus générale reste possible (notamment par un processus stochastique [66]), cela implique généralement un temps de résolution de l'ordre de plusieurs heures par image, ce qui pour de nombreuses applications, dont le rendu de séquences animées, reste prohibitif. Nous allons présenter ici quelques hypothèses classiques permettant de simplifier le problème posé et d'aboutir à des algorithmes de résolution plus efficaces.

Dans le cas général, le terme d'émission locale $L_{créée}^\lambda(x, \omega^\uparrow)$ devrait prendre en compte le phénomène de fluorescence (*i.e.* la réémission de l'énergie lumineuse absorbée à une certaine longueur d'onde dans une autre gamme de fréquence). Dans ce cas il n'est pas possible de résoudre l'équation 1.7 séparément pour les différentes longueurs d'onde. Glassner a donné dans [41] une forme générale de l'équation du rendu permettant de prendre en compte la fluorescence. Toutefois, même si le phénomène est très fréquemment observé (il joue un rôle très important notamment dans l'aspect du papier, ou des peintures blanches), il reste très rarement simulé. On supposera dans le reste de ce mémoire que les équations pour les différentes longueurs d'ondes sont découplées. Afin d'alléger les notations, nous ne précisons plus explicitement la nature monochromatique des grandeurs utilisées.

Pour ramener la simulation de l'éclairage global à un problème de taille raisonnable, Goral *et al.*[44] ont proposé, s'inspirant d'approches classiques en ingénierie thermique ([107, 109]), de ne considérer que les échanges purement diffus entre surfaces opaques. Ainsi, la *BSDF* de toutes les surfaces ² est constante pour toutes les directions de S^+ et nulle pour les autres. On supposera de même que toutes les sources de lumière sont des émetteurs purement diffus. Cette hypothèse est centrale dans les méthodes dites *de radiosité* (*cf.* paragraphe 1.2). Rappelons que l'équation 1.1 définit la radiosité comme l'intégrale sur toutes les directions de S^+ de la luminance sortant en un point par unité de surface. Nous pouvons de même remplacer tous les termes directionnels de l'équation 1.7 par des grandeurs hémisphériques :

$$\begin{cases} \forall (\omega^\downarrow, \omega^\uparrow) \in S^{+2}, & \rho_{bds}(x, \omega^\downarrow, \omega^\uparrow) = \frac{\rho(x)}{\pi} \\ \forall \omega^\uparrow \in S^+, & L_{créée}(x, \omega^\uparrow) = \frac{E(x)}{\pi} \\ \forall \omega^\uparrow \in S^+, & L(x, \omega^\uparrow) = \frac{B(x)}{\pi} \end{cases}$$

On obtient donc l'équation de la radiosité :

$$B(x) = E(x) + \rho(x) \int_{\mathcal{M}} B(y) \frac{\cos \theta \cos \theta'}{\pi \|x - y\|^2} v(x, y) dy \quad (1.8)$$

Étude formelle des équations

Les équations 1.7 et 1.8 appartiennent à la famille des équations de Fredholm de 2^e espèce. Étant données une fonction g et une fonction k dite *fonction noyau*, une forme générale des problèmes de cette classe est la recherche de la fonction f telle que :

$$f(x) = g(x) + \int k(x, y) f(y) dy \quad (1.9)$$

Si l'on définit l'opérateur K tel que

$$(Kf)(x) = \int k(x, y) f(y) dy \quad (1.10)$$

alors on peut réécrire la forme générale des équations de Fredholm de 2^e espèce (1.9) de la façon suivante :

$$f = g + Kf$$

²En toute rigueur, il s'agit ici d'une *BRDF*

Si l'on se place dans un espace de fonctions approprié (comme par exemple $\mathcal{L}_2(\mathcal{M})$), on notera que l'opérateur K est linéaire. De plus, il existe³ une solution unique à cette équation $f = (I - K)^{-1}g$. On sait que dans le cas où la norme de l'opérateur $\|K\| \equiv \sup_{f \neq 0} \frac{\|Kf\|}{\|f\|}$ est inférieure à 1, l'opérateur $(I - K)$ est nécessairement inversible et son inverse est donné par la série de Neumann :

$$(I - K)^{-1} = \sum_{j=0}^{\infty} K^j \quad (1.11)$$

Cette condition est vérifiée dans le cas des équations de l'illumination globale. En effet l'opérateur K décrit comment la lumière incidente en chaque point de la scène contribue par réflexion ou transmission au flux lumineux quittant ce point. Le fait que sa norme soit strictement inférieure à 1 signifie qu'il ne peut y avoir autant ou plus d'énergie redistribuée que d'énergie incidente. L'équation 1.11 peut être interprétée de la façon suivante : l'énergie lumineuse présente dans la scène à l'équilibre est la somme de l'énergie directement émise par les sources lumineuses (donnée par la fonction $L_{crée}$), et de celle obtenue après j réflexions/transmissions, j variant de 1 à l'infini.

Les équations 1.7 et 1.8 possèdent donc une unique solution. De plus, nous pouvons obtenir une suite d'approximations de plus en plus proches de cette solution par application successive de l'itération $f_{n+1} = Kf_n + g$ (la fonction initiale f_0 étant indifférente, elle peut être choisie identiquement nulle⁴). En effet [6] :

$$\|f - f_n\| \leq \frac{\|K\|^{n+1}}{1 - \|K\|} \|g\|$$

Ainsi cette méthode convergera d'autant plus vite si l'énergie initialement émise dans la scène ($\|g\|$) est faible et si seulement une faible partie de la lumière est réémise à chaque interaction (*i.e.* si $\|K\|$ est faible). Ce résultat est conforme à l'intuition puisque tronquer la série 1.11 au rang j revient à négliger les chemins lumineux faisant intervenir plus de j réflexions et/ou transmissions.

1.2 La méthode de radiosit 

La m thode de radiosit , classique en  tude des  changes thermiques, a  t  pour la premi re fois appliqu e   la synth se d'images par Goral *et al.* [44]. C'est une m thode de r solution de l' quation 1.8 appartenant   la famille des m thodes d' l ments finis [63, 61, 62]. Ces derni res visent   r soudre num riquement une  quation int grale en cherchant, dans un espace de fonction de dimension finie, une fonction la plus proche possible de la solution. Pour ce faire, l' quation int grale originelle est projet e dans le dit espace de fonctions, afin de ramener le probl me pos    la r solution d'un syst me d' quations lin aire.

Malheureusement, m me simplifi  de telle mani re, le calcul d'une solution reste d'un co t prohibitif pour une sc ne de taille respectable, puisque il faut calculer pour chaque paire d' l ments de la sc ne un ou plusieurs coefficients d'interaction (les coefficients du syst me lin aire   inverser), faisant intervenir la fonction de visibilit  v , particuli rement co teuse    valuer [64, 114]. Nous pr senterons dans les paragraphes suivants quelques unes des am liorations successivement apport es   l'algorithme original afin d'en am liorer l'efficacit , et dont nous nous sommes inspir s pour d velopper notre algorithme d'illumination globale pour les animations que nous pr senterons dans les deux parties suivantes de ce m moire.

³si une telle inversion est d finie, ce qui n'est pas  vident si nous raisonnons dans un espace vectoriel de dimension infinie

⁴Un choix «  clair  » de f_0 plus proche de la solution recherch e r duira le temps de convergence de la solution.

1.2.1 Discrétisation

Principe

Nous supposons par la suite que la fonction E appartient à un espace fonctionnel $\mathcal{F}(\mathcal{M})$ défini de la façon suivante [24] :

$$\mathcal{F}(\mathcal{M}) = \{f : \mathcal{M} \rightarrow \mathbb{R}, f \in (\mathcal{L}_2(\mathcal{M}) \cap \mathcal{L}_\infty(\mathcal{M}))\}$$

Cette fonction représentant une puissance lumineuse par unité de surface, il est en effet naturel de la supposer finie (donc appartenant à $\mathcal{L}_\infty(\mathcal{M})$). De plus, nous aurons besoin par la suite des notions d'orthogonalité et de produit scalaire (définis dans $\mathcal{L}_2(\mathcal{M})$). Ainsi, $\mathcal{F}(\mathcal{M})$ est un espace de Banach au sens des normes $\|\cdot\|_1$ et $\|\cdot\|_\infty$ et un espace de Hilbert au sens de la norme $\|\cdot\|_2$ associée au produit scalaire ⁵.

$$\langle f, g \rangle = \int_{\mathcal{M}} f(x)g(x) dx$$

On supposera de plus que l'opérateur K (cf. équation 1.10) est de norme strictement inférieure à 1 au sens des trois normes d'opérateur associées. Ainsi, nous sommes assurés de l'existence d'une *unique* solution B appartenant elle-même à $\mathcal{F}(\mathcal{M})$ (cf. paragraphe 1.1.2).

Toutefois, cet espace étant de dimension infinie, il nous est impossible de représenter de façon exacte ses éléments en mémoire (sauf à résoudre formellement l'équation 1.8 ce qui n'est pas possible dans le cas général). Le processus de *discrétisation* permet, dans les méthodes d'éléments finis, de passer d'une équation intégrale initiale à un problème proche dans un sous-espace $\mathcal{F}_N(\mathcal{M})$ de dimension finie N plus aisément manipulable.

Soit un ensemble de fonctions $(u_i)_{1 \leq i \leq N}$ formant une famille orthogonale dans cet espace. Nous allons chercher dans ce sous-espace une fonction \tilde{B} minimisant le résidu :

$$r(x) = \tilde{B}(x) - E(x) - (K\tilde{B})(x) \quad (1.12)$$

Cette minimisation peut être définie de plusieurs façons. On peut, par exemple, souhaiter annuler ce résidu en une série de points particuliers (ce que l'on désigne sous le nom de *méthode de collocation*). Nous préférons ici la méthode de Galerkin, pour laquelle le résidu doit être orthogonal à l'espace $\mathcal{F}_N(\mathcal{M})$. On notera bien que \tilde{B} n'est pas la fonction la plus proche de B dans $\mathcal{F}_N(\mathcal{M})$ (i.e. son projeté orthogonal B^\perp) mais la fonction qui « vérifie le mieux » l'équation 1.8, ce qu'illustre la figure 1.4.

De même que pour B dans $\mathcal{F}(\mathcal{M})$, l'existence et l'unicité de \tilde{B} est assurée, et de plus, il y a convergence forte de la suite des solutions vers B lorsque N tend vers l'infini [76].

Le résidu devant être orthogonal à tous les u_i , nous obtenons un système linéaire :

$$\forall i \in [1, N] \quad \langle r, u_i \rangle = \langle \tilde{B}, u_i \rangle - \langle E, u_i \rangle - \langle K\tilde{B}, u_i \rangle = 0$$

Or \tilde{B} peut s'exprimer comme une combinaison linéaire des fonctions de base u_j :

$$\tilde{B} = \sum_{j=1}^N B_j u_j$$

donc

$$\forall i \in [1, N] \quad \left[\sum_{j=1}^N (\langle u_i, u_j \rangle - \langle K u_j, u_i \rangle) B_j \right] = \langle E, u_i \rangle$$

⁵Sauf précision du contraire, c'est à cette norme que l'on fera référence par la suite

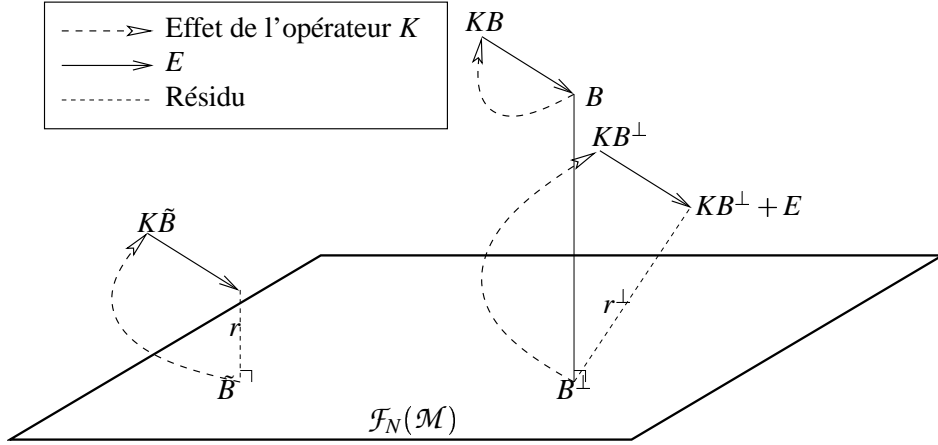


FIG. 1.4: Distinction entre B , B^\perp , \tilde{B} : La fonction B est l'exacte solution de l'équation 1.8. B^\perp est la fonction la plus proche de cette solution dans $\mathcal{F}_N(\mathcal{M})$. Elle vérifie mal cette équation. \tilde{B} , solution du problème discrétisé, est la fonction de $\mathcal{F}_N(\mathcal{M})$ qui vérifie le mieux l'équation de la radiosité.

Si l'on note :

$$\forall (i, j) \in [1, N]^2 \quad M_{i,j} = \frac{\langle Ku_j, u_i \rangle}{\|u_i\|^2}$$

et

$$\forall i \in [1, N] \quad E_i = \frac{\langle E, u_i \rangle}{\|u_i\|^2}$$

on peut reformuler le système linéaire précédent sous la forme d'une équation matricielle :

$$\begin{pmatrix} 1 - \frac{\langle Ku_1, u_1 \rangle}{\|u_1\|^2} & -\frac{\langle Ku_2, u_1 \rangle}{\|u_1\|^2} & \dots & -\frac{\langle Ku_N, u_1 \rangle}{\|u_1\|^2} \\ -\frac{\langle Ku_1, u_2 \rangle}{\|u_2\|^2} & 1 - \frac{\langle Ku_2, u_2 \rangle}{\|u_2\|^2} & \dots & -\frac{\langle Ku_N, u_2 \rangle}{\|u_2\|^2} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\langle Ku_1, u_N \rangle}{\|u_N\|^2} & -\frac{\langle Ku_2, u_N \rangle}{\|u_N\|^2} & \dots & 1 - \frac{\langle Ku_N, u_N \rangle}{\|u_N\|^2} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{pmatrix} = \begin{pmatrix} \frac{\langle E, u_1 \rangle}{\|u_1\|^2} \\ \frac{\langle E, u_2 \rangle}{\|u_2\|^2} \\ \vdots \\ \frac{\langle E, u_N \rangle}{\|u_N\|^2} \end{pmatrix} \quad (1.13)$$

Résolution

La méthode de radiosité consiste donc en deux étapes :

1. Le calcul des coefficients du système linéaire
2. La résolution du système linéaire

Les différentes variantes de cette méthode proposées dans la littérature peuvent différer sur les points suivants :

- Le choix des fonctions de base u_i (fonctions uniformes par morceaux [44], polynômes d'ordre supérieur [121], ondelettes [46, 96], à support sur un maillage régulier, de densité adaptée aux variations de B [22], ou aligné sur les discontinuités de la fonction v [59, 70], etc.)
- La méthode d'évaluation des coefficients $M_{i,j}$
- Le choix d'une méthode de résolution du système linéaire

L'algorithme original de Goral *et al.* [44] utilise des fonctions uniformes par morceaux définies sur un maillage régulier prédéterminé, calcule les coefficients de couplage $M_{i,j}$ en utilisant le théorème de Stokes sur les intégrales de contours et en supposant la visibilité totale, et procède à la résolution du système linéaire par la méthode du pivot. En conséquence, il ne s'adresse qu'à des configurations très particulières de scènes, et ne permet pas d'effectuer des calculs sur un maillage très fin.

Bases d'éléments constants, facteurs de forme

Dans le cas particulier des éléments constants, la base employée est généralement composée des fonctions indicatrices de chacun des éléments de maillage P_i d'aire $A_i = \|u_i\|^2$. De plus, on suppose généralement que la réflectance est constante sur chaque P_i . Ainsi on voit apparaître dans la matrice M des coefficients $F_{i,j}$ nommés *facteurs de forme* :

$$\forall (i, j) \in [1, N]^2, M_{i,j} = \rho_i F_{i,j}$$

avec :

$$\forall (i, j) \in [1, N]^2, F_{i,j} = \frac{1}{\pi A_i} \int_{P_i} \int_{P_j} \frac{\cos \theta \cos \theta'}{\|x - y\|^2} v(x, y) dy dx \quad (1.14)$$

Les facteurs de forme sont des grandeurs purement géométriques, s'exprimant en unités inverses d'un angle solide. On peut démontrer [98] que ces facteurs de forme vérifient un certain nombre de relations algébriques intéressantes :

$$\begin{aligned} \forall i \in [1, N], \quad \sum_{j=1}^N F_{i,j} &\leq 1 \\ \forall (i, j) \in [1, N]^2, \quad A_i F_{i,j} &= A_j F_{j,i} \\ \forall (i, j) \in [1, N]^2, \quad \text{si } P_j &= P_{j'} \cup P_{j''}, \text{ alors } F_{i,j} = F_{i,j'} + F_{i,j''} \end{aligned} \quad (1.15)$$

Comme leur calcul est une des principales difficultés de l'algorithme de radiosit , de nombreux articles y sont consacr s, en thermique [108, 86], mais aussi dans le domaine de la synth se d'images, o  ont  t  propos es aussi bien des m thodes analytiques [9], que des techniques employant le mat riel graphique [23]. On trouvera en particulier un catalogue de diff rentes formules analytiques pour des configurations particuli res dans [65] et un  tat de l'art des diff rentes m thodes de calcul dans [89].

1.2.2 Radiosit  progressive

Il est possible de r soudre num riquement le syst me d' quations 1.13 en employant des m thodes it ratives classiques (Jacobi, Gauss-Seidel, *etc.*) [43]. Le temps de r solution et la place m moire n cessaire pour ces m thodes varient comme le carr  de N , nombre d' l ments formant le maillage. Parmi ces m thodes de r solution, la m thode de relaxation de Southwell pr sente un int r t particulier. Elle admet une interpr tation physique simple :   chaque it ration, un  l ment est choisi et sa radiosit  est  mise vers l'ensemble des surfaces de la sc ne. On choisit g n ralement l' l ment poss dant la plus grande radiosit  non encore r  mise. Ainsi, il est possible d'obtenir d s les premi res it rations une sc ne int gralement  clair e.

Bien que cette m thode converge  galement en $O(n^2)$, si l'on accepte de recalculer les facteurs de forme n cessaires   chaque it ration, il n'est plus n cessaire de m moriser qu'une seule ligne de la matrice M pour une it ration. Son co t m moire varie donc comme une fonction lin aire du nombre d' l ments. De plus, il est dans ce cas possible d'affiner le maillage   chaque it ration pour s'adapter aux variations locales de la solution. Les m thodes s'inspirant de ce principe sont appel es m thodes de radiosit  progressive [22].

1.2.3 Radiosité hiérarchique

Principe

L'utilisation de structures de données hiérarchiques est une approche classique en algorithmique. On souhaite, par une factorisation bien choisie des données, obtenir une accélération substantielle du traitement. La problématique est ici de choisir quel niveau de représentation est le plus approprié pour formuler les échanges énergétiques entre les différents objets de la scène. Par exemple, dans le cas de la figure 1.5, la radiosité du livre S_1 sur la commode contribuera plus à l'éclairage de la surface P qu'à celui du livre sur l'étagère S_3 . De même, il n'est pas forcément nécessaire de connaître le contour des ombres portées sur un mur pour calculer sa contribution à l'éclairage du mur opposé.

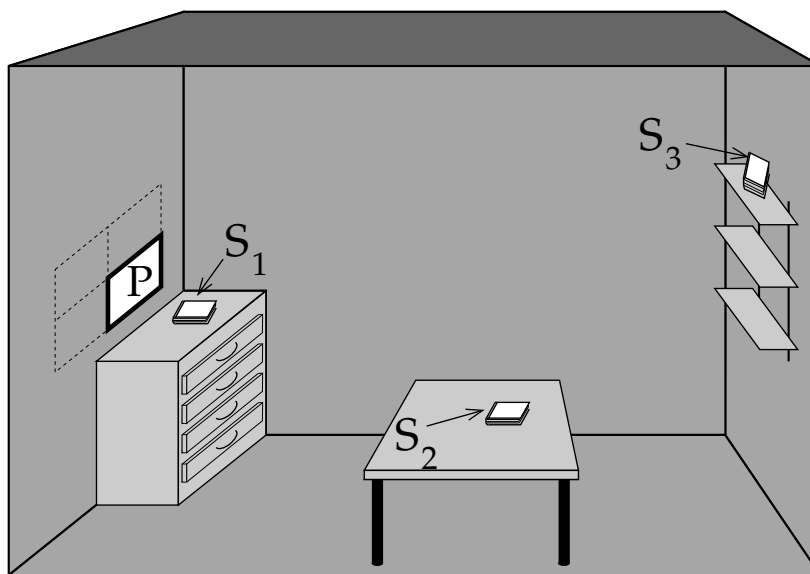


FIG. 1.5: Diversité de l'importance des échanges⁶

S'inspirant de méthodes développées en mécanique pour résoudre des problèmes d'interactions gravitationnelles entre plusieurs corps, Hanrahan *et al.* ont proposé un algorithme hiérarchique de résolution de l'équation matricielle 1.13 [55]. Ils proposent, pour chaque surface composant la scène, de créer un maillage hiérarchique (par exemple en utilisant un schéma de construction récursif de type *arbre quaternaire*). Afin d'indiquer à quel niveau de la hiérarchie se produit tel ou tel échange, des liens sont établis entre les éléments (contenant, entre autres informations, le facteur de forme).

Le processus récursif de placement des liens dans la hiérarchie permet de créer par effet de bord un maillage adapté à la fonction de radiosité : si le lien actuellement considéré ne satisfait pas un certain critère (permettant de déterminer s'il est correctement placé) alors on subdivise émetteur ou récepteur, et on examine la possibilité d'établir des liens au niveau hiérarchique obtenu.

Il existe différents critères possibles pour l'établissement de liens. Il s'agit en général d'estimer la quantité d'énergie qui transitera à ce niveau de la hiérarchie (comme suggéré par Hanrahan *et al.*), ou encore l'erreur que produira cette factorisation particulière [69].

⁶Cette figure, ainsi que les figures 1.6 et 1.7, sont extraites du livre de François Sillion et Claude Puech *Radiosity and Global Illumination* [104]

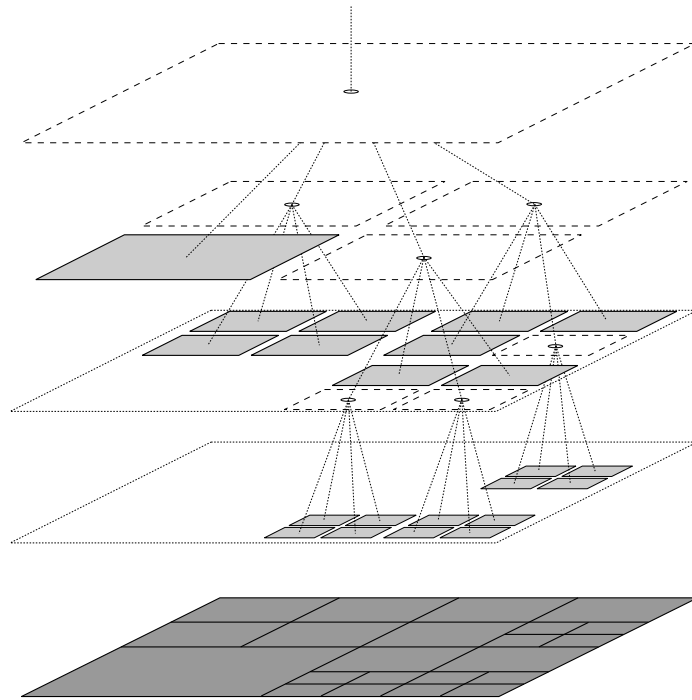


FIG. 1.6: Hiérarchie utilisant un arbre quaternaire

Sous réserve d'un choix « correct » du critère de raffinement, Hanrahan *et al.* ont postulé que le nombre de liens créés par élément de surface est à peu près constant. Dans ce cas, le coût asymptotique de l'algorithme est de l'ordre de $O(N + k^2)$ où N est le nombre d'éléments de surface et k est le nombre de surfaces originel (il faut en effet dans un premier temps lier toutes les surfaces vers toutes les autres surfaces, ce qui peut prendre un temps non-négligeable de l'ordre de $O(k^2)$ [64]).

Solution du problème hiérarchique

Le calcul de radiosité hiérarchique est donc un processus itératif, chaque itération étant composée de trois passes :

1. **Le raffinement des liens**
2. **La collecte des irradiances** (*gather*) : on calcule grâce aux liens l'irradiance reçue par chaque élément
3. **La maintenance de la cohérence hiérarchique** (*push-pull*, cf. figure 1.7) : on effectue un parcours récursif de la hiérarchie en descendant l'irradiance de chaque élément à ses fils (ajout de l'irradiance à celle de ses fils) et en remontant la radiosité (moyenne pondérée par les aires des radiosités des fils). Le calcul de la radiosité à partir de l'irradiance reçue se fait alors sur les feuilles de la hiérarchie.

On note :

- \mathbb{E} : Ensemble des éléments de la hiérarchie
- \mathbb{S} ($\mathbb{S} \subset \mathbb{E}$) : Ensemble des sources de lumière
- \mathbb{L}_e ($\forall e \in \mathbb{E}$) : Ensemble des éléments liés à l'élément e
- I_e^i ($\forall e \in \mathbb{E}$) : Irradiance reçue par l'élément e à l'itération i
- B_e^i ($\forall e \in \mathbb{E}$) : Radiosité de l'élément e à l'itération i .
- ρ_e ($\forall e \in \mathbb{E}$) : Réflectance de l'élément e
- $F_{e,l}$: Facteur de forme de l'extrémité du lien l vers l'élément e .

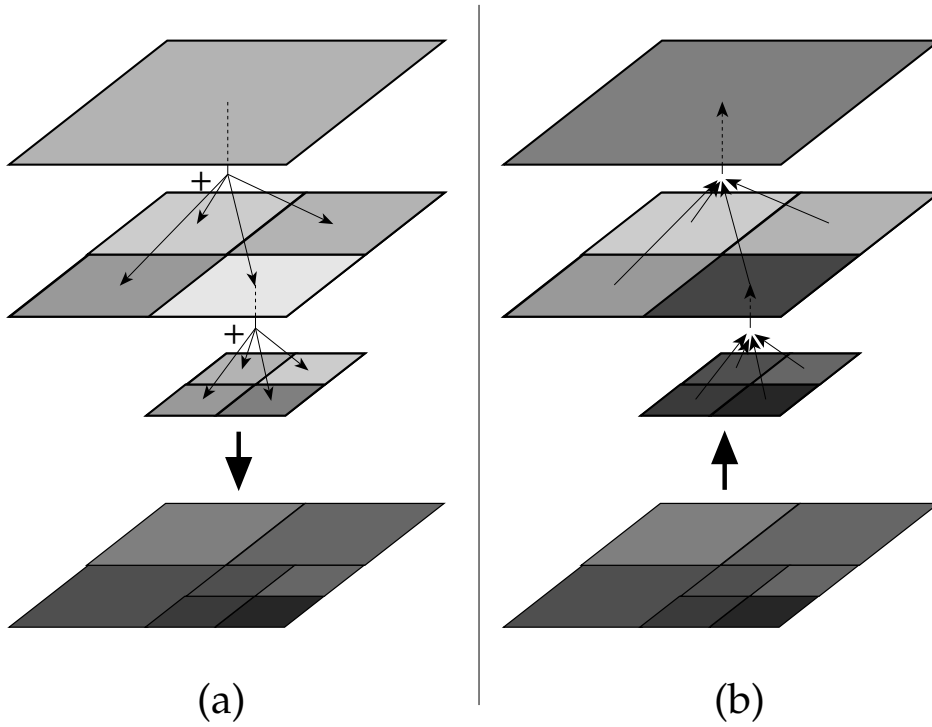


FIG. 1.7: Mise à jour hiérarchique

Cet algorithme se base sur deux variables pour chaque élément de la hiérarchie : I_e et B_e . Toute la radiosité est recalculée à chaque itération pour chaque élément. Il est à noter qu'avant la première itération, seules les sources de lumière possèdent une radiosité non nulle ($\forall e \in \mathbb{E}, B_e^0 \neq 0 \iff e \in \mathbb{S}$).

Cette précision étant faite, voici l'algorithme pour l'étape i :

1. **Raffinement** : $\forall e \in \mathbb{E}, \forall l \in \mathbb{L}_e$, faire descendre l jusqu'à un niveau hiérarchique où il vérifie le critère de raffinement.
2. **Collecte** : Pour tous les éléments de la scène, calcul de la contribution de chaque lien.

$$\forall e \in \mathbb{E}, I_e^i = \sum_{l \in \mathbb{L}_e} F_{e,l} B_l^{i-1}$$

3. **Mise à jour** : En descendant la hiérarchie, on somme les irradiances.

$$\forall e \in \mathbb{E}, I_e^i = I_e^i + I_{\text{père}(e)}^i$$

4. Sur les feuilles f de la hiérarchie : Calcul de la radiosité

$$B_f^i = \rho_f I_f^i$$

5. **Mise à jour** : En remontant la hiérarchie, la radiosité du père devient la moyenne de celle de ses fils

$$\forall e \in \mathbb{E}, B_e^i = \frac{\sum_{k \in \text{fils}(e)} \text{Aire}(k) B_k^i}{\text{Aire}(e)}$$

1.2.4 Regroupement des surfaces

L'algorithme de radiosité hiérarchique permet de réduire considérablement la complexité des calculs de radiosité. Toutefois, le coût en varie toujours quadratiquement par rapport au nombre de surfaces composant la scène en entrée. Ainsi, pour les scènes constituées d'un nombre important de polygones (par exemple celles issues de modeleurs procédant à une triangulation fine des surfaces), le coût de la phase initiale d'établissement des liens au plus haut niveau de la hiérarchie devient très important [64], puisqu'elle implique que pour chaque paire de surfaces, il faut évaluer le critère de raffinement, qui nécessite généralement un calcul de facteur de forme ou au moins une estimation de la visibilité mutuelle des dites surfaces.

Afin de permettre une utilisation plus optimale de l'algorithme hiérarchique, il est donc nécessaire d'étendre la hiérarchie *au-dessus* des surfaces, en créant un nouveau type d'éléments hiérarchiques permettant le regroupement des différents objets de la scène (*clustering*) [105, 99]. L'étape préliminaire d'établissement des liens sera ainsi remplacée par l'établissement d'un unique lien de l'objet regroupant toutes les surfaces de la scène vers lui-même. Les cas d'obstruction triviale (par exemple pour des surfaces se trouvant dans des pièces différentes dans le cas de calculs d'éclairage en intérieur) seront détectés à haut niveau et rejetés en groupe par le processus de raffinement (*cf.* figure 1.8).



FIG. 1.8: Exemple d'occlusion triviale : les objets situés au rez-de-chaussée ne peuvent interagir directement avec ceux situés à l'étage. L'utilisation d'une hiérarchie de groupes permet de détecter de tels cas au début du processus de raffinement des liens, permettant ainsi un gain de temps substantiel.

Cela soulève deux problèmes distincts :

- La construction de la hiérarchie des groupes au-dessus des surfaces
- Le calcul des interactions entre groupes et entre groupes et surfaces

Construction de la hiérarchie des groupes

Les mécanismes de regroupement hiérarchiques de surfaces pour la radiosité sont en général inspirés des méthodes de construction de structures d'accélération pour le tracé de rayon, comme par exemple par subdivision récursive de cubes (*octree* [39] ou arbres binaires), ou par construction récursive de boîte englobantes [42, 21]. La hiérarchie de groupes est par ailleurs généralement également utilisée comme structure d'accélération des requêtes de visibilité, notamment pendant les calculs de facteur de forme ou lors de l'évaluation du critère de raffinement. Le lecteur trouvera une étude comparative de quelques unes des principales méthodes de construction de hiérarchies de groupes pour la radiosité hiérarchique dans [56] ainsi qu'en annexe de ce document.

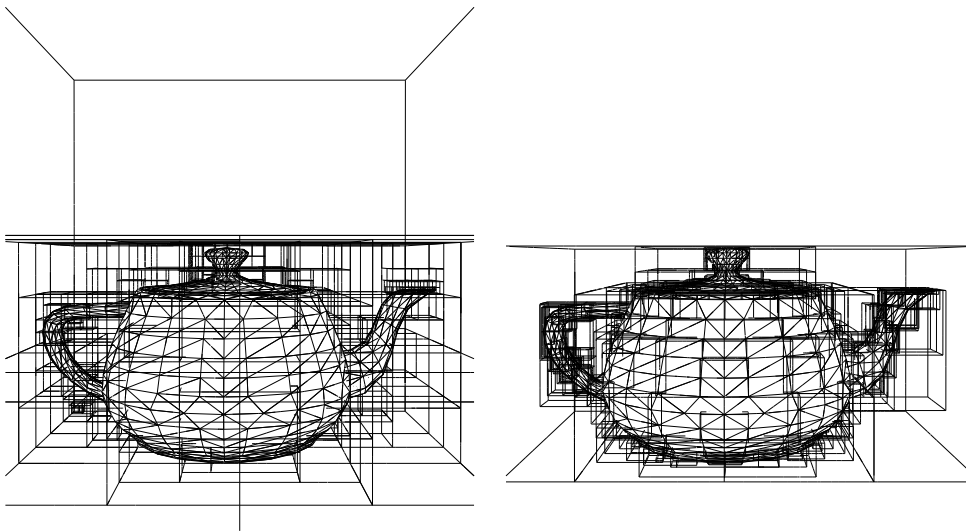


FIG. 1.9: Exemples de hiérarchies de groupes : à gauche : arbre binaire, à droite : boîtes englobantes

On obtient ainsi une hiérarchie mixte d'objets, dans laquelle un groupe peut avoir pour fils d'autres groupes aussi bien que des surfaces, alors que les surfaces ne peuvent elles-mêmes n'avoir pour descendants que d'autres surfaces.

Interactions entre groupes ou avec un groupe

Afin de ne pas perdre tout le bénéfice de la formulation hiérarchique, l'évaluation du critère de raffinement pour une interaction impliquant un ou plusieurs groupes doit être largement plus rapide que la somme des évaluations impliquant toutes les surfaces contenues dans ce groupe. En revanche, cette estimation ne doit pas être trop grossière pour que le processus de raffinement puisse placer correctement les liens dans la hiérarchie (de plus, une estimation trop hâtive de la visibilité pourrait également soit ne pas détecter suffisamment tôt une occlusion totale, soit, ce qui est plus grave, provoquer la suppression d'un lien entre deux groupes dont deux surfaces sont pourtant mutuellement visibles, causant ainsi une sous-estimation importante de la radiosité).

De même, lors de la collecte de l'énergie à travers les liens (*cf.* paragraphe 1.2.3), il faut pouvoir évaluer rapidement et avec une précision raisonnable la quantité d'énergie reçue ou émise par un groupe de surfaces. On notera, entre autres, que l'on ne peut pas supposer que les groupes sont des émetteurs isotropes, ou que la contribution à l'irradiance des surfaces contenues dans un groupe due à un lien arrivant plus haut dans la hiérarchie est uniforme (*cf.* figure 1.10).

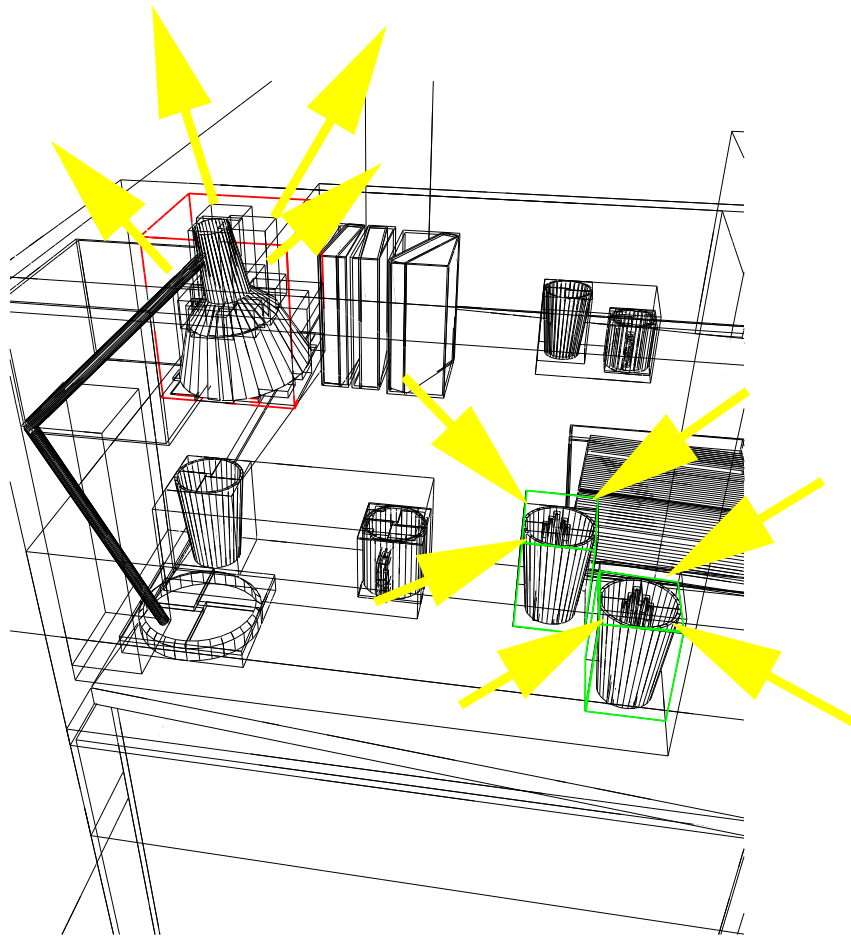


FIG. 1.10: Anisotropie des groupes : Si l'on considère les groupes **en vert** comme des récepteurs diffus, alors la lumière provenant de la lampe leur sera transmise comme provenant de toutes les directions. De même si le groupe **en rouge** contenant la lampe est considéré comme un émetteur isotrope, alors de la lumière sera émise de façon incorrecte dans toutes les directions (notamment vers le haut).

Or, l'essentiel du coût de l'évaluation du facteur de forme entre éléments provient de l'évaluation de la visibilité. Le temps passé à l'évaluation des termes angulaires de la forme « $\cos(\theta) \times \cos(\theta')$ » est négligeable en comparaison. De ce fait, Smits *et al.* [105] et Sillion [99] suggèrent de n'utiliser les groupes que pour factoriser les calculs de visibilité, tout en effectuant l'évaluation des termes géométriques au niveau des surfaces. En pratique, il nous faut distinguer deux cas :

- **En émission** : l'irradiance reçue en un point x par un élément quelconque en provenance d'un groupe est égale à la somme des contributions de chaque surface Q_i que ce groupe contient.

$$I(x) = \sum_i \int_{Q_i} B_{Q_i} \frac{\mathcal{R}(x) \cos \theta'}{\|x - y\|^2} dy$$

\mathcal{R} étant le facteur de réception défini dans [99] comme valant $\cos \theta$ si le récepteur est une surface et 1 s'il s'agit d'un groupe (l'orientation des surfaces sera prise en compte plus tard). Cette irradiance sera ensuite modulée par un terme de visibilité dont le coût d'évaluation doit être indépendant du nombre de surfaces contenues dans le groupe (on peut, par exemple, échantillonner la visibilité sur une grille de points fixes).

- **En réception** : au moment de la phase de collecte des irradiances à travers la structure de liens, il faut pour les groupes distribuer *immédiatement* l'irradiance reçue à travers un lien donné à toutes les surfaces contenues dans ce groupe. L'irradiance reçue par une surface P_i sera donc $I_{P_i} = I_{groupe} \cos \theta_i$. La nécessité d'effectuer immédiatement cette distribution provient du fait que l'on ne souhaite pas stocker chaque contribution avec sa direction de provenance.

1.2.5 Utilisation d'autres bases de fonctions

Bien que la plupart des algorithmes dérivés de la méthode de radiosité n'emploie que des bases de fonctions uniformes par morceaux, un certain nombre de solutions employant des bases de fonctions plus sophistiquées ont été proposées dans la littérature. L'utilisation de fonctions de base d'ordre plus élevé dans les méthodes d'éléments finis permet d'améliorer la vitesse de convergence (lorsqu'on raffine le maillage) des solutions discrètes vers la solution théorique. En effet, comme le montre la figure 1.11, on peut attendre de l'approximation d'une fonction qu'elle soit d'autant plus satisfaisante que l'ordre des fonctions de base est plus élevé.

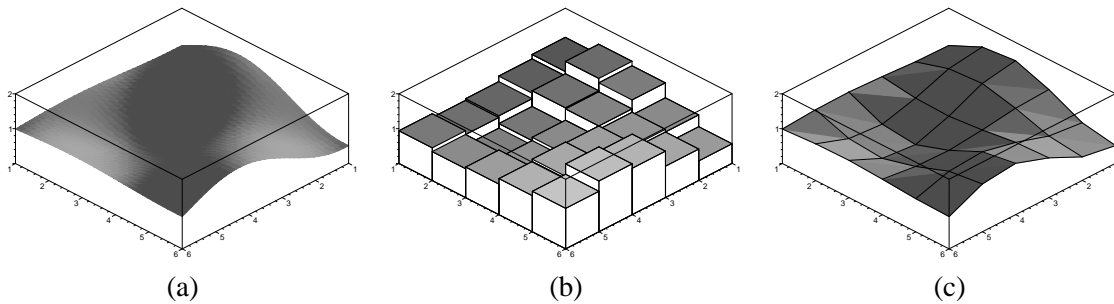


FIG. 1.11: Approximation d'une fonction quelconque (a) par une fonction uniforme par morceaux (b) et par une fonction linéaire par morceaux (c)

En général, les fonctions de bases dans \mathbb{R}^2 sont construites en effectuant le produit tensoriel de deux bases de fonctions de \mathbb{R} . La figure 1.12 illustre ce procédé de construction.

Dans [115], Troutman *et al.* procèdent à une comparaison de la méthode de Galerkin et de la méthode par collocation pour la discrétisation de l'équation de radiosité et proposent deux algorithmes utilisant ces méthodes avec des fonctions de bases polynomiales par morceaux. Dans [121], Zatz montre comment construire une base de fonction orthonormale à partir des polynômes de Legendre (figure 1.13) et de Jacobi. De plus, afin de remédier aux problèmes d'intégration liés aux discontinuités de la fonction de visibilité, il propose de découpler le calcul de la projection de celui de la visibilité en intégrant séparément cette dernière afin de calculer un masque de visibilité (*shadow mask*).

Gortler *et al.* montrent, dans [46], que la formulation hiérarchique de la méthode de radiosité n'est qu'un cas particulier d'une famille de méthodes numériques basées sur la théorie des ondelettes [52, 35, 77, 71] (on trouvera également une introduction aux applications de la théorie des ondelettes pour la synthèse d'images dans [112]). Ils y montrent qu'à raffinement équivalent, l'utilisation de bases d'ondelettes (multi-ondelettes de Alpert [4], ainsi que leur propre base d'ondelettes uniformes par morceaux baptisées *flatlets*) permet d'améliorer considérablement la qualité des solutions calculées.

Malheureusement, l'utilisation d'une base d'ondelettes a eu pour contrepartie, jusque récemment, un coût mémoire important. En effet, une base de multi-ondelettes d'ordre n en dimension 2 est composée de n^2 fonctions (puisque cette base est formée des produits tensoriels des n fonctions de base correspondantes en dimension 1). Lorsque l'on procède à la projection de l'équation de radiosité sur cette base, on retrouve la notion de « facteur de forme » sous la forme d'une matrice de coefficients d'interaction, de largeur égale au nombre de fonctions de base employées. Ainsi, pour chaque paire

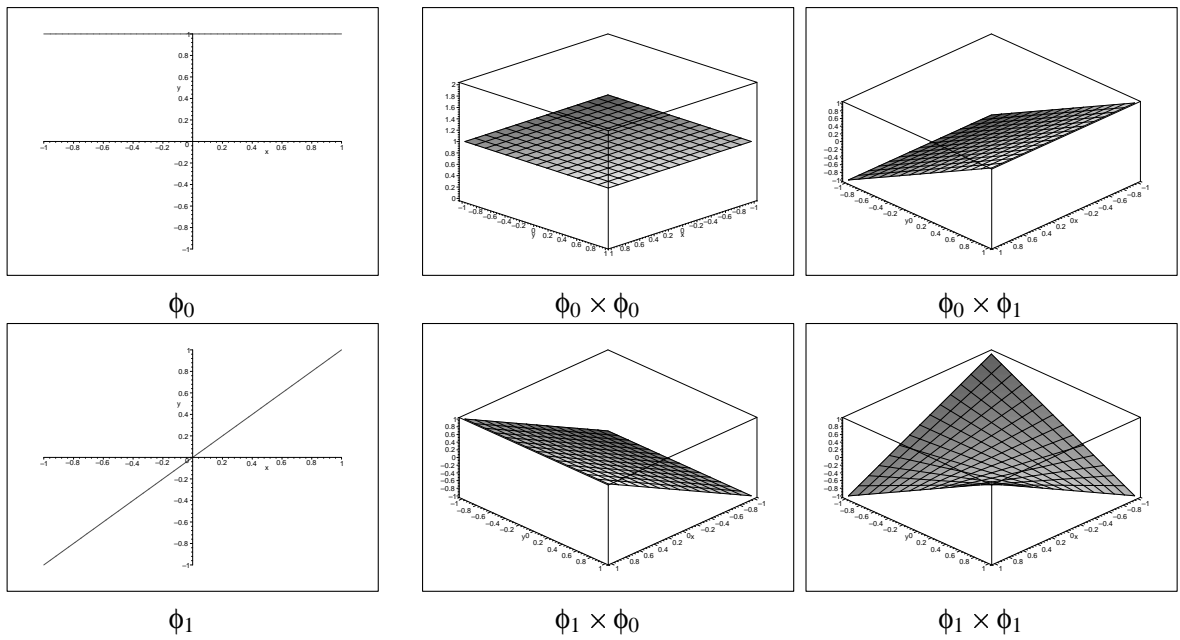


FIG. 1.12: Construction d'une base de fonctions sur \mathbb{R}^2 à partir d'une base de fonctions sur \mathbb{R}

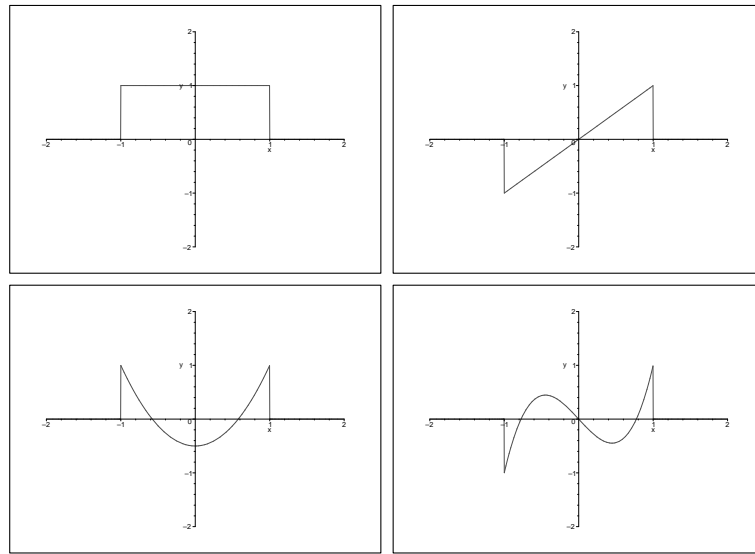


FIG. 1.13: Polynômes orthogonaux de Legendre de degré 0, 1, 2, et 3

d'éléments de la scène, n^4 coefficients sont nécessaires pour décrire l'interaction. Si l'on considère le fait qu'aucune des implémentations de la radiosité hiérarchique avec ondelettes n'utilise à ce jour à notre connaissance le regroupement des surfaces, on voit que le coût mémoire en est prohibitif (pour une scène originellement composée de 1000 surfaces, *avant même tout raffinement*, le stockage en mémoire de la hiérarchie des liens requiera donc au moins 16 millions de flottants dans le cas d'ondelettes linéaires, 81 millions dans le cas d'ondelettes quadratiques). En conséquence, l'étude expérimentale de Willmott *et al.* [120] conclut que les bases d'ondelettes d'ordre élevés sont, en pratique, rendues inutilisables par leur consommation mémoire pour une utilisation sur des scènes de taille importante.

Récemment, Stamminger *et al.* [111] et Granier *et al.* [49] ont montré qu'en procédant à l'émission de l'énergie (*shooting*) plutôt qu'à sa récolte (*gathering*), on pouvait implémenter l'algorithme de radiosité hiérarchique en évitant le stockage coûteux des liens. En appliquant ces techniques et en se basant sur le développement de nouveaux oracles de raffinement par Bekaert *et al.* [11, 12] et Gibson *et al.* [38], Cuny *et al.* [25] montrent que les bases de multi-ondelettes peuvent être avantageusement utilisées au sein de l'algorithme de radiosité hiérarchique, pour des scènes de taille conséquente, avec un coût mémoire raisonnable.

Indépendamment des considérations de précision de la simulation, un avantage important des bases de fonctions d'ordre supérieur sur les bases de fonctions uniformes est qu'elles simplifient grandement le problème de l'affichage de la solution. En effet, dans ce cas, on peut se passer des méthodes de reconstruction employées pour obtenir des images d'apparence continue (*i.e.* où le maillage n'est pas apparent, *cf.* paragraphe 1.2.6) une fois le calcul de la solution effectué. En particulier, l'affichage des éléments lorsque l'on emploie des ondelettes linéaires, se fait très simplement en utilisant les facilités offertes par la plupart des cartes graphiques disponibles à ce jour (lissage de Gouraud [47]).

1.2.6 Affichage de la solution

La méthode de radiosité peut être employée pour effectuer des simulations d'éclairage, notamment pour le *design* architectural, afin de prévoir l'impact de différents paramètres sur la luminosité d'un bâtiment. Pour de telles simulations, l'apparence de la solution n'importe pas tant que sa précision. On peut dans de tels cas afficher directement la solution, en utilisant le matériel graphique. Pour chaque élément de maillage, on affichera un polygone ayant pour couleur une fonction de sa radiosité.

En revanche, lorsque l'on souhaite utiliser cette méthode à des fins esthétiques, le résultat brut obtenu n'est généralement pas acceptable. Si l'on a employé une base de fonctions uniformes pour représenter la radiosité, à moins d'effectuer un raffinement extrêmement fin, le maillage sous-jacent apparaîtra de façon flagrante.

La méthode la plus simple pour remédier à ce problème tente de reconstruire une fonction linéaire par morceaux à partir de la solution de radiosité. Pour ce faire, il suffit d'affecter à chaque sommet du maillage une radiosité calculée en moyennant les radiosité des éléments environnants. Toutefois, l'utilisation de l'interpolation de Gouraud peut entraîner des discontinuités dans l'affichage dans le cas de polygones concaves, ou lorsqu'un point n'est sommet que d'un côté de la frontière entre deux éléments (*T-vertex*). Il est nécessaire, pour traiter ces discontinuités, de modifier le maillage après le calcul de la solution [92, 8].

Toutefois, cette méthode de reconstruction ne permet de faire disparaître les discontinuités de la fonction de radiosité que lorsqu'elles interviennent au niveau le plus fin de la hiérarchie. Celles résultant, par exemple, d'une précision insuffisante dans le calcul de l'éclairage indirect resteront apparentes. De plus, cette reconstruction est équivalente à l'application d'un filtre sur la solution. Elle n'améliore pas réellement sa précision mais au contraire répartit les erreurs entre les éléments.

D'autres méthodes plus coûteuses ont été proposées pour produire à partir d'une solution de radiosité une image de qualité [103, 90]. Ces méthodes, dites de « récolte finale », utilisent la solution de radiosité produite pour générer une image dépendante du point de vue par lancer de rayons (*cf.* figure 1.14). La radiosité en chaque point de l'image est calculée en évaluant les facteurs de formes du point vers chacun des éléments liés à l'élément auquel il appartient. Cette passe de récolte finale est généralement beaucoup plus longue à effectuer que la résolution du problème hiérarchique (les mesures effectuées dans [93] indiquent que le calcul de la solution de radiosité hiérarchique ne représente généralement que 10% du temps de production total d'une image).



FIG. 1.14: Exemples de résultats de *récoltes finales* (images calculées par Xavier Granier)

Quelle que soit l'approche adoptée, il faut également décider comment obtenir une couleur à partir de valeurs de radiosité générées pour différentes longueurs d'onde. En pratique, de nombreuses implémentations de l'algorithme de radiosité emploient seulement trois longueurs d'onde différentes, qui seront mises directement en correspondance avec les composantes rouge, vert et bleu de l'espace des couleurs du matériel graphique. Il reste à déterminer comment adapter des puissances lumineuses dont l'amplitude des variations est de l'ordre de 10^7 aux dispositifs d'affichage des ordinateurs, capables seulement de représenter une gamme d'amplitudes de l'ordre de 10^2 . Une mise en correspondance linéaire peut produire des images beaucoup trop sombres ou des zones de saturation à proximité des sources de lumière. Dans la littérature, des solutions ont été proposées afin de prendre en compte la luminosité globale de l'image [119], ou des critères perceptuels pour produire des images présentant de forts contrastes [85].

Illumination pour les animations

PUISQUE la mémoire dont un ordinateur dispose est finie, il est plus aisé de développer un algorithme manipulant des données dans un espace discret que de raisonner dans un espace continu, où l'on sera condamné à une approche formelle. La représentation d'une image en mémoire se fait donc généralement sous la forme d'un échantillonnage du plan sur une grille de points (ou *pixels*). De même une animation est généralement représentée en mémoire comme une suite discrète d'images. On procède donc à un échantillonnage de la dimension temporelle, généralement à intervalles réguliers suffisamment brefs pour créer l'illusion du mouvement (25 images par seconde étant le standard hérité du cinéma). De ce fait, pour calculer une séquence d'animation, la plus simple méthode possible est d'exécuter un algorithme quelconque de synthèse d'images fixes autant de fois que l'animation désirée devra compter d'images, en remettant à jour avant chaque exécution les informations nécessaires (géométrie à l'instant considéré, propriétés des matériaux, *etc.*).

Toutefois, le temps de calcul d'une seule image de synthèse reste très important, et ce malgré l'impressionnante progression des performances des ordinateurs et des algorithmes de synthèse au cours des vingt dernières années. Un grand nombre de telles images est nécessaire à la production d'un plan même très court d'un film d'animation. Il paraît donc évident que l'on peut attendre du développement d'approches spécifiques aux animations une réduction substantielle des coûts de production dus au temps de calcul.

De nombreux algorithmes ont été proposés se basant sur la grande similitude des calculs effectués pour plusieurs images consécutives. Nous ferons référence par la suite à cette propriété de similitude en tant que *cohérence temporelle*. Elle provient du fait que la fréquence d'échantillonnage temporel des animations est généralement très grande en comparaison de la vitesse de variation des phénomènes simulés (mouvements des objets, changements de propriétés optiques des surfaces, *etc.*). On peut donc s'attendre de façon globale à de faibles variations des grandeurs calculées d'une image à l'autre, même si de brusques changements restent possibles localement (par exemple à cause des occultations dues à un objet en mouvement)¹.

¹C'est pourquoi le terme de « cohérence » nous paraît préférable à celui de « continuité ».

On peut formellement distinguer deux types de cohérence pouvant être exploitées. En effet, on peut tirer profit de la cohérence temporelle du résultat final, *i.e.* du peu de différences entre deux images consécutives, ou *cohérence dans l'espace image*. Mais il est également possible de tirer profit de l'uniformité temporelle des résultats intermédiaires, avant projection sur la caméra, c'est-à-dire la *cohérence dans l'espace objet*. Nous nous intéresserons par la suite plus particulièrement à cette dernière forme de cohérence, car elle nous paraît plus prometteuse pour la simulation de l'illumination globale.

Dans ce chapitre, nous montrerons dans un premier temps que cette cohérence est suffisamment importante pour justifier le développement d'algorithmes cherchant à en tirer profit. Nous montrerons ensuite qu'au delà des considérations sur la vitesse d'exécution, l'utilisation pour un calcul image par image de certains algorithmes d'illumination globale non spécifiquement prévus pour les animations peut provoquer des incohérences temporelles indésirables. Finalement, nous terminerons ce chapitre par un rapide tour d'horizon des méthodes proposées à ce jour pour l'exploitation de la cohérence temporelle dans les calculs d'illumination globale.

2.1 Propriétés de l'irradiance vue comme une fonction du temps

2.1.1 Une brève étude des variations d'éclairement au cours d'une animation

Nous avons choisi deux scènes de test présentant des caractéristiques d'éclairage différentes pour étudier l'importance de la cohérence temporelle de l'illumination globale entre les différentes images d'une animation. Nous n'avons pas envisagé les variations d'intensité des sources de lumière (cas qui peut être traité par d'autres moyens [37]), nous nous sommes en revanche focalisés sur les changements induits par le déplacement des objets.

La première scène de test (*cf.* figure 2.1, à gauche), à laquelle nous nous référerons par la suite sous le nom de ACTEURS, modélise une pièce dans laquelle deux objets sont en mouvement. La taille et la trajectoire de ces objets ont été choisis de façon à représenter un déplacement possible de deux acteurs dans une pièce, pour anticiper sur l'application éventuelle de notre technique à l'intégration d'images de synthèse dans un film. La seconde scène de test (*cf.* figure 2.1, à droite), que nous baptiserons ÉCRAN nous montre une pièce similaire dans laquelle les conditions d'éclairement changent grandement pour la quasi-totalité des objets au cours de la séquence, à cause d'un écran en mouvement faisant obstacle à l'une des sources de lumière. Ces animations durent toutes deux 3 secondes (75 images).

Pour pouvoir estimer le temps machine occupé à effectuer des calculs redondants lorsque l'on veut faire le rendu d'une animation complète image par image, nous avons calculé une solution de radiosité séparée par image en sauvegardant à chaque fois la hiérarchie complète des liens. Nous comparons alors les hiérarchies de liens obtenues et calculons la « durée de vie » de chaque lien, c'est-à-dire le nombre d'images pendant lequel il est resté identique (nous considérons deux liens comme identiques s'ils joignent les deux mêmes éléments et s'ils transportent la même quantité de lumière).

Cela nous permet de calculer, pour n'importe quelle durée, la quantité d'énergie totale qui a été échangée à travers des liens ayant une telle durée de vie. Le tableau 2.1 indique la proportion d'énergie (pour l'éclairement direct et indirect) qui s'est révélée échangée de façon constante par deux éléments *tout au long* de l'animation (*i.e.* passant par des liens qui sont restés identiques pour toutes les images). Le détail de la quantité moyenne d'énergie en fonction de la durée durant laquelle elle est restée constante est donné dans la figure 2.2.

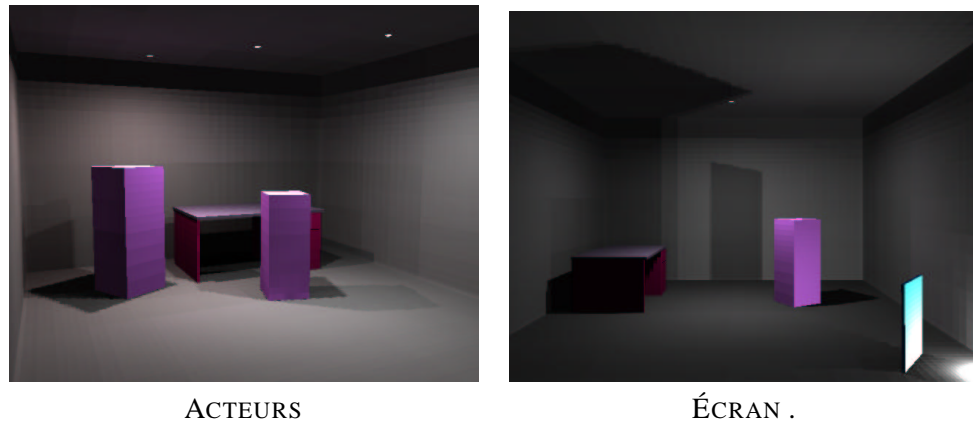


FIG. 2.1: Deux scènes de test choisies pour illustrer des situations pouvant être rencontrées dans des applications pratiques. ÉCRAN présente des variations d'éclairage plus évidentes et à plus grande échelle (aussi bien en temps qu'en espace) dues au mouvement de l'obstacle devant la source de lumière située à droite au niveau du sol.

	Éclairage direct	Éclairage indirect
ACTEURS	69.1%	49.5%
ÉCRAN	63.9%	42.7%

TAB. 2.1: Proportion d'énergie constante durant toute l'animation

Nous pouvons voir qu'une grande partie de l'énergie échangée dans la scène n'a besoin d'être calculée qu'une seule fois. Ceci est essentiellement dû aux liens partant des sources de lumière vers les parties non dissimulées des murs et des objets statiques (le bureau) pour l'éclairage direct, et entre les murs pour l'éclairage indirect. Le fait qu'une quantité significative de liens peut être considérée comme valide pendant plus d'une image mais pas pendant l'animation entière rend une approche multi-résolution comme la radiosité hiérarchique particulièrement intéressante. On notera que dans la scène ÉCRAN, une quantité d'énergie significative est transportée par des liens qui ne sont valides que pour une seule image, et ce en raison des importants changements dans l'éclairage ayant lieu au cours de l'animation.

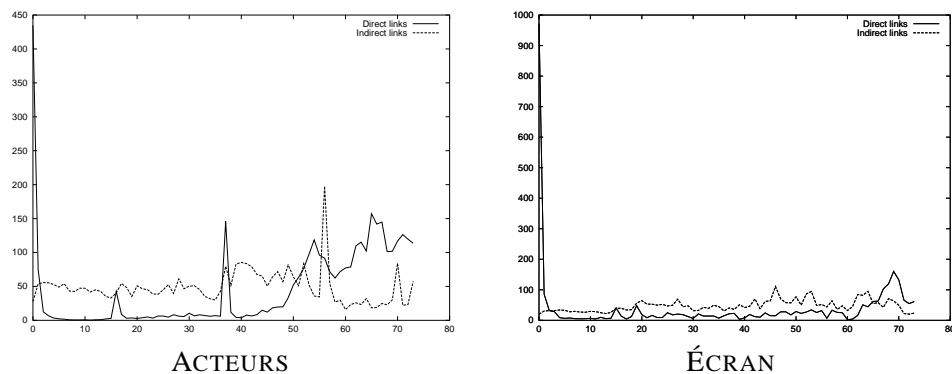


FIG. 2.2: Moyenne de l'énergie lumineuse émise en fonction du nombre d'images durant lequel le lien correspondant est resté valide (unités arbitraires). Pour des raisons d'échelle, ces figures ne prennent pas en compte les liens qui sont restés valides durant toute l'animation (se reporter au tableau 2.1)

2.1.2 Continuité temporelle de l'éclairage

Nous avons montré dans la section précédente que la cohérence temporelle de l'éclairage était suffisante pour que l'on puisse espérer en tirer partie pour accélérer les calculs d'illumination globale pour les animations. Toutefois, il nous faut prendre garde à ce que les algorithmes développés à cette fin ne diminuent pas la qualité des animations obtenues. Un calcul imprécis peut en effet introduire des défauts dans les images produites, mais également dégrader la continuité temporelle de l'animation. En effet, de brusques changements dans les conditions d'éclairage d'un objet (disparition soudaine d'une ombre, changement de teinte intempestive, *etc.*) sont très rapidement repérés par l'oeil lorsque l'on visionne l'animation, alors qu'ils seront évidemment invisibles si on observe les images séparément, en dehors de toute séquence.

De tels défauts peuvent être provoqués par des effets de seuil dans les calculs d'éclairage. Par exemple, on peut souhaiter employer l'algorithme de radiosité hiérarchique que nous avons décrit au paragraphe 1.2.3 pour calculer une série d'images destinées à la production d'une animation. Dans ce cas, l'utilisation d'oracles de raffinement basé sur un critère basé sur une mesure liée à la quantité d'énergie retransmise par ce lien peut provoquer localement des discontinuités *temporelles* dans l'éclairage. Supposons que l'énergie transmise d'un objet vers un mur soit en constante augmentation sur un intervalle de temps donné, partant d'une quantité négligeable puis devenant de plus en plus importante (par exemple dans le cas où cet objet se rapproche du mur). Dans le cas d'un calcul image par image, il se produira probablement une discontinuité lorsque l'illumination indirecte franchira le seuil de raffinement (*cf.* figure 2.3).

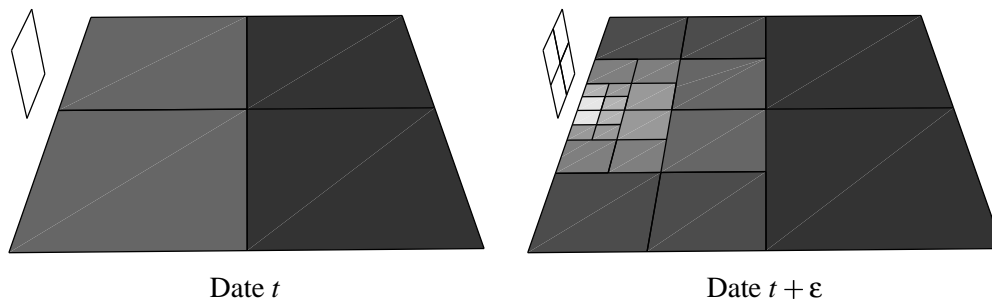


FIG. 2.3: Discontinuité temporelle due à un effet de seuil : Un léger mouvement de la surface émettrice vers la droite fait que le critère de subdivision passe un seuil et provoque un raffinement en cascade. Cela se traduit par une brusque hausse de la radiosité en certains points de l'élément récepteur.

Ce type de problème n'est pas spécifique aux seuls calculs image par image. En effet, ils peuvent apparaître avec n'importe quel algorithme procédant incrémentalement, sans connaissance (ou sans tenir compte) de ce qui se produira après l'image actuellement calculée (nous rappellerons le principe de ces méthodes au paragraphe 2.2.1).

De même, des défauts suffisamment faibles pour être acceptés sur une image fixe deviennent intolérables dans une animation s'ils ne sont pas cohérents dans le temps. Ainsi, il est difficile à ce jour d'employer les méthodes stochastiques de calcul de l'éclairage global (tracé de chemins, méthode de Metropolis, lancer de photons, *etc.*) pour le rendu des images d'une animation car le bruit inhérent à ces méthodes provoque une sorte de scintillement dans l'espace image (similaires aux parasites analogiques apparaissant sur un téléviseur mal réglé). Afin de faire disparaître ce bruit, il est nécessaire soit d'envisager un filtrage temporel des images [78], soit de rendre le processus d'échantillonnage aléatoire spatial cohérent dans le temps [15]. On notera que des problèmes similaires sont rencontrés lorsque l'on désire générer une animation dans un style non-photoréaliste, en garantissant une certaine cohérence temporelle aux coups de pinceau ou de crayon [75, 87].

2.2 Méthodes classiques pour la simulation de l'éclairage dans les animations

Nous allons dans ce paragraphe faire un tour d'horizon des algorithmes qui ont été proposés dans le domaine de la synthèse d'images spécifiquement pour le calcul de l'éclairage de scènes animées. Pour plus de détails, le lecteur intéressé trouvera dans [113] un état de l'art des méthodes d'accélération du rendu des animations antérieures à 1994. Il pourra également trouver dans la thèse de Orti [83] une présentation des différentes extensions de la méthode de radiosit  au cas des scènes animées.

Divers paramètres influençant l'éclairage de la scène sont susceptibles de varier au cours d'une animation. D'une part, il est possible de jouer sur la puissance des sources de lumière, ce qui modifie globalement la quantité d'énergie circulant dans la scène, et d'autre part les objets peuvent être déplacés, ce qui implique une modification de l'opérateur décrivant la redistribution de l'énergie après réflexion.

Le traitement du premier type de changement est beaucoup plus aisé. En effet, d'après l'équation 1.11, on sait que la solution de l'équation du rendu [66], lorsqu'elle existe, est une fonction linéaire des « conditions initiales » du problème, c'est-à-dire de la fonction décrivant la puissance d'émission des sources de lumière. Après avoir calculé quelques solutions avec des conditions d'éclairage linéairement indépendantes, on peut donc très rapidement obtenir toutes les images de l'animation par combinaison linéaire des premières images obtenues. Dans le cas général, déterminer le nombre d'images de référence nécessaire et calculer les coefficients de la combinaison linéaire peuvent s'avérer difficile. Dans le cas particulier des algorithmes de radiosit , l'espace des fonctions dans lequel on recherche une solution étant de dimension finie, il est en revanche aisé d'obtenir des solutions exactes par interpolation : en effet, il suffit de calculer une image par source de lumière et par bande de fréquence souhaitée, où cette source sera le seul émetteur [2, 29] (*cf.* figure 2.4) On disposera ainsi de l'image de chacune des fonctions de base de l'espace des solutions, ce qui détermine intégralement l'opérateur linéaire K .

Malheureusement, il est beaucoup moins aisé de déterminer l'influence sur la solution d'une modification apportée à la géométrie de la scène. En effet, de telles modifications nécessitent généralement de recalculer les interactions entre les surfaces en mouvement et celles restées statiques, et également de réévaluer la visibilité entre un plus grand nombre de surfaces encore. Nous nous intéresserons donc par la suite essentiellement aux algorithmes ayant été proposés pour effectuer de façon efficace ces calculs.

Ces méthodes peuvent être groupées en deux catégories :

- Les méthodes *dynamiques* ou *séquentielles*, qui, à partir d'une image calculée, calculent la série des images suivantes en cherchant à se restreindre aux seules parties de la solution nécessitant une modification. Se destinant à des applications interactives, ces méthodes ne peuvent se baser sur la connaissances d'évènements situés en aval dans le temps.
- Les méthodes *statiques* ou à *accès temporel arbitraire*, pour lesquelles on suppose disposer de la connaissance complète de la séquence des évènements et des trajectoires. Ces méthodes ne peuvent donc effectuer de calculs en temps-réel puisqu'elles nécessitent de pouvoir accéder de façon arbitraire à n'importe quelle date de l'animation.

On notera que, puisque leurs hypothèses de fonctionnement sont moins restrictives, les méthodes dynamiques peuvent parfaitement être employées en lieu et place de n'importe quelle méthode statique. Toutefois, ces algorithmes ne peuvent que réagir aux changements de configuration dans la scène qu'ils ne peuvent anticiper. De plus, ces algorithmes sont généralement employés dans des

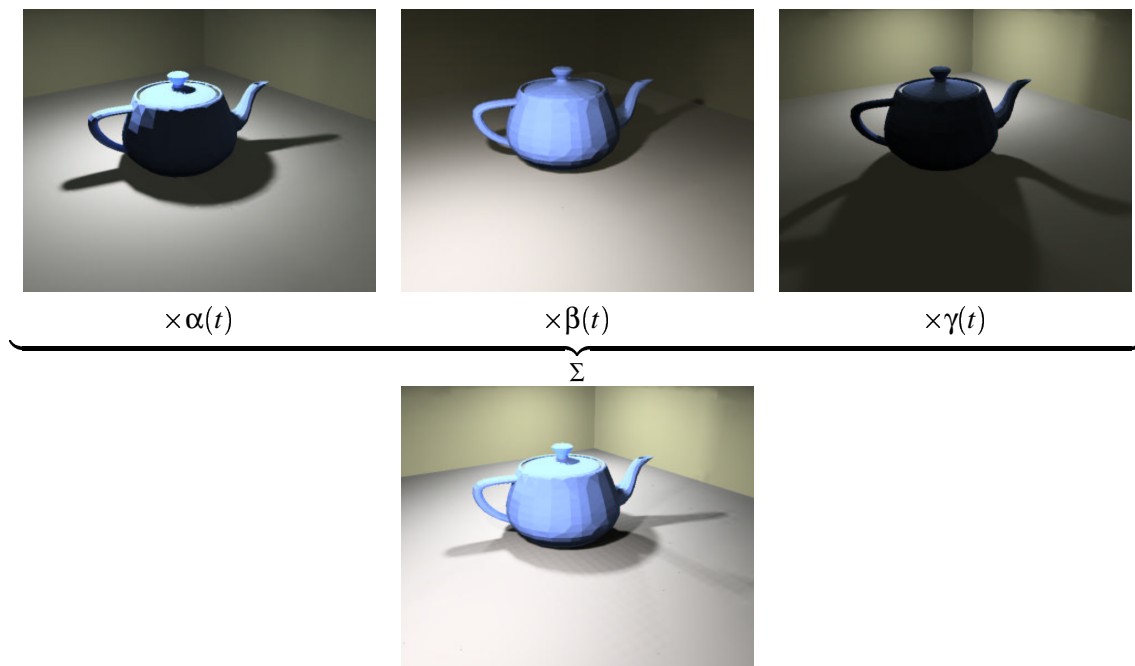


FIG. 2.4: Ré-éclairage interactif d'une scène : Dans le cas où seule l'intensité des sources de lumière varie, on peut calculer chaque image de l'animation par combinaison linéaire des images obtenues en activant chacune des sources séparément.

applications destinées à une utilisation interactive. Les approximations qu'ils effectuent, afin de permettre une meilleure fréquence de mise à jour, ne leur permettent pas d'obtenir des résultats de qualité équivalente à ceux d'un algorithme qui ne serait pas contraint au respect d'une certaine cadence de production d'images.

2.2.1 Méthodes dynamiques

Méthodes de rendu asynchrones

Relativement peu d'algorithmes ont été proposés pour la mise à jour interactive de solutions calculées par une méthode dérivée du tracé de rayons. Chaque point de l'image étant calculé et affiché séparément, ces algorithmes nécessitent en effet un travail plus important pour localiser et recalculer les zones nécessitant modification.

Se basant sur les travaux de Bergman *et al.* [13], Bishop *et al.* [16] proposent de mettre à jour les points de l'écran dans un ordre aléatoire au fur et à mesure que l'algorithme de tracé de rayons les produit. En ne cherchant pas à synchroniser la mise à jour de l'image avec la fréquence de balayage de l'écran, cet algorithme rend possible une mise à jour des images aussi rapide que le permettent les ressources de calcul à disposition. Suivant un principe similaire, Walter *et al.* [118] proposent une méthode permettant de découpler le moteur de rendu du processus de mise à jour des points de l'image. Il leur est ainsi possible de naviguer ou d'interagir avec un modèle en utilisant un algorithme de lancer de rayons ou de tracé de chemins [66]. Les points calculés pour les images précédentes peuvent être réutilisés par re-projection. Différentes priorités sont affectées aux zones de l'écran afin d'orienter le tirage aléatoire des échantillons.

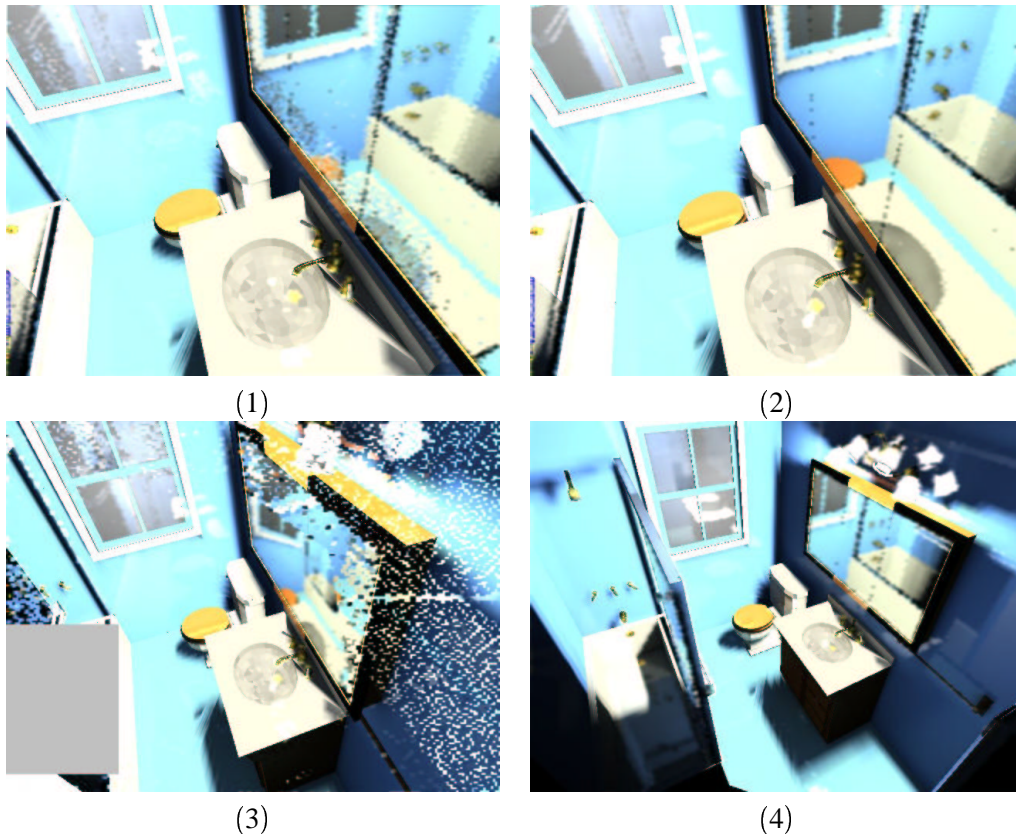


FIG. 2.5: Mise à jour de l'affichage du "cache de rendu" [118] lors d'un mouvement de caméra ³. Les points sont progressivement recalculés si la caméra cesse son déplacement.

Ces algorithmes permettent une interaction de l'utilisateur avec la scène dans des conditions acceptables. Toutefois, pour pouvoir permettre un temps de réaction raisonnable, ils ne garantissent pas une qualité d'image consistante au cours du temps (*cf.* figure 2.5). Destinés plus spécifiquement aux phases préliminaires de construction et de visualisation d'un modèle 3D, ils ne peuvent réellement être employés pour la génération d'animations de synthèse de qualité.

Radiosité incrémentale

En revanche, les méthodes de radiosité se prêtent plus naturellement au développement d'algorithmes interactifs. Elles permettent en effet le calcul d'une solution indépendante du point de vue. Un déplacement de l'observateur ne nécessite aucun calcul d'éclairage supplémentaire. Seul l'affichage doit être ré-effectué. Cette phase ne requiert habituellement que le tracé des polygones formant le maillage et peut donc bénéficier des facilités offertes par la plupart des cartes graphiques actuellement sur le marché. La difficulté provient donc uniquement de la détermination de l'impact sur l'éclairage des modifications apportées à la géométrie. En effet, cela implique le recalcul d'un nombre important de facteurs de forme.

Chen [19] et George *et al.* [37] ont présenté la même année deux algorithmes relativement similaires basés sur le principe de la radiosité progressive, permettant d'apporter une réponse en un temps raisonnable à une modification de la scène, et ce sans nécessairement pousser les calculs jusqu'à

³Ces images ont été calculées à l'aide du logiciel GIS de Granier [48].

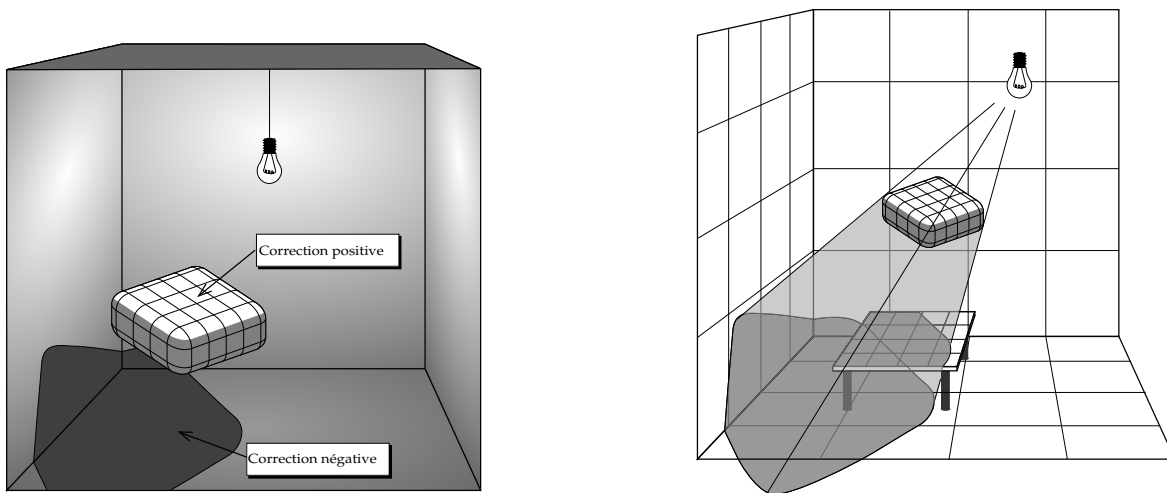


FIG. 2.6: Correction de l'énergie due à un objet en mouvement [19, 37]⁵ : **à gauche** : corrections à apporter à la radiosité due au mouvement d'un objet dynamique, **à droite** : utilisation d'un volume d'ombre pour déterminer les éléments devant être mis à jour.

convergence. Le déplacement d'un objet est traité de la façon suivante : la radiosité des surfaces le composant est d'abord ré-émise vers toutes les autres surfaces de la scène. Puis, un élément de surface fixe est choisi, sa radiosité est re-propagée vers l'objet en mouvement, et l'opposé de sa radiosité est propagé vers les surfaces contenues dans l'ombre de l'objet déplacé (déterminées par l'emploi d'un volume d'ombre [82]). On choisit alors un second élément et on répète le même processus, jusqu'à ce qu'il n'y ait plus d'éléments nécessitant une re-propagation d'énergie. Afin de permettre à ces modifications d'avoir lieu alors même que les déplacements d'objets sont encore en cours, les opérations restant à effectuer sont mémorisées dans une file d'attente et effectuées au rythme que permet la charge de calcul. Ce procédé a permis à Chen d'obtenir un facteur d'accélération de 1,2 à 3,6 par rapport à un calcul image par image dans le cas de modifications apportées à la géométrie.

Radiosité hiérarchique dynamique

L'algorithme de radiosité hiérarchique [55], tel que nous l'avons présenté au paragraphe 1.2.3, se prête naturellement à une extension au cas des scènes dynamiques. En effet, la structure des liens employée par cet algorithme peut être vue comme une représentation multi-résolution de l'opérateur de redistribution de la lumière K . Elle permet un accès rapide aux facteurs de forme potentiellement modifiés par le mouvement d'un objet.

Forsyth *et al.* [34] ont proposé la première adaptation de l'algorithme de radiosité hiérarchique au cas des scènes animées. Après le calcul d'une solution de radiosité correspondant à la configuration géométrique initiale, la hiérarchie des liens est progressivement mise à jour conformément aux mouvements d'un objet. Ainsi, à chaque nouvelle image à produire, un lien peut être occulté ou validé (en cas de passage de l'objet dynamique entre les deux éléments reliés), promu ou descendu dans la hiérarchie (si le mouvement d'une des deux extrémités fait que le niveau de précision auquel le dit lien se situe n'est plus suffisant ou devient excessif, *cf.* figure 2.7). Afin de limiter la quantité de liens à réévaluer, les changements de visibilité pour les éléments statiques sont ignorés, et les facteurs de forme pour les liens impliquant une surface en mouvement ne sont calculés exactement que pour quelques étapes clés et obtenus par interpolation le reste du temps.

⁵Ces deux figures sont extraites du livre de Sillion *et al.* *Radiosity and Global Illumination* [104]

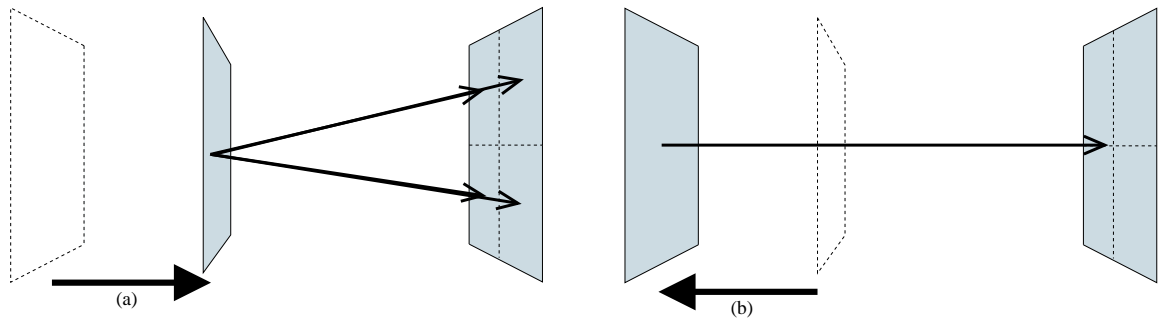


FIG. 2.7: Mise à jour dynamique d'un lien [34] : Après déplacement de la surface réceptrice (a), le lien représenté est jugé insuffisamment précis et est donc raffiné. En cas de retour à la situation de départ (b), les quatre liens obtenus seront remplacés par le lien originel. Toutefois, le maillage restera raffiné.

Malheureusement, le mécanisme de mise à jour des liens de [34], s'il permet d'améliorer ou de réduire le niveau de précision d'un lien, ne prévoit en revanche que d'augmenter la précision du maillage, et ne le simplifie jamais, ce qu'illustre la figure 2.7. La taille du maillage de la scène augmente donc rapidement. Pour remédier à ce problème, Shaw propose dans [97] une structure de liens alternative permettant de corriger le raffinement du maillage au fur et à mesure de l'animation. Cette structure permet de « dé-raffiner » le maillage après le passage d'une ombre ou après la promotion d'un lien dans la hiérarchie. De plus, afin de repérer rapidement les interactions occultées par le passage d'un objet dynamique, elle ne considère que les liens interceptant le volume englobant la trajectoire de l'objet dynamique.

Drettakis *et al.* [30], adjoignent à la hiérarchie des liens des faisceaux [53] reliant les boîtes englobantes des deux extrémités de chaque lien (*cf.* figure 2.8). Ainsi, utilisant un algorithme de radiosité hiérarchique avec regroupement [105, 99], ils opèrent une traversée hiérarchique de l'espace des droites pour repérer les faisceaux interceptés par la boîte englobante de l'objet en mouvement. Les facteurs de forme à recalculer, ainsi que les modifications à opérer sur le maillage sont ainsi rapidement identifiés. De plus cette méthode permet, afin de garantir une fréquence de mise à jour compatible avec une utilisation interactive, de limiter la profondeur de la descente dans la hiérarchie. Schoeffel *et al.* [94], afin de réduire le coût mémoire engendré par le stockage de la hiérarchie des liens et des faisceaux correspondants, utilisent un mécanisme permettant de créer les faisceaux à la volée en utilisant un algorithme de prédiction de trajectoire. Granier *et al.* [50] ajoutent à l'algorithme de [30] la possibilité de traiter les réflexions non diffuses. Leur algorithme incrémental combine radiosité hiérarchique pour le calcul des interactions diffuses et méthode de lancer de particules utilisant des textures pour l'affichage des caustiques. Afin de permettre l'affichage des réflexions spéculaires en direction de la caméra (qui font que l'aspect de certains objets est dépendant du point de vue), ils emploient le mécanisme de cache de [118].

2.2.2 Méthodes statiques

Rendu dépendant du point de vue

Glassner [40] propose une variante de l'algorithme du tracé de rayons où les calculs d'intersection sont effectués dans un espace à quatre dimensions. Ils utilisent en particulier une généralisation de la hiérarchie de boîtes englobantes classiques employant des hypervolumes 4D. Ceci permet d'obtenir un facteur d'accélération de l'ordre de 2. Besuievsky *et al.* [15, 14] proposent pour calculer l'éclairage pour une animation d'employer une méthode de résolution stochastique sur une scène plus complexe

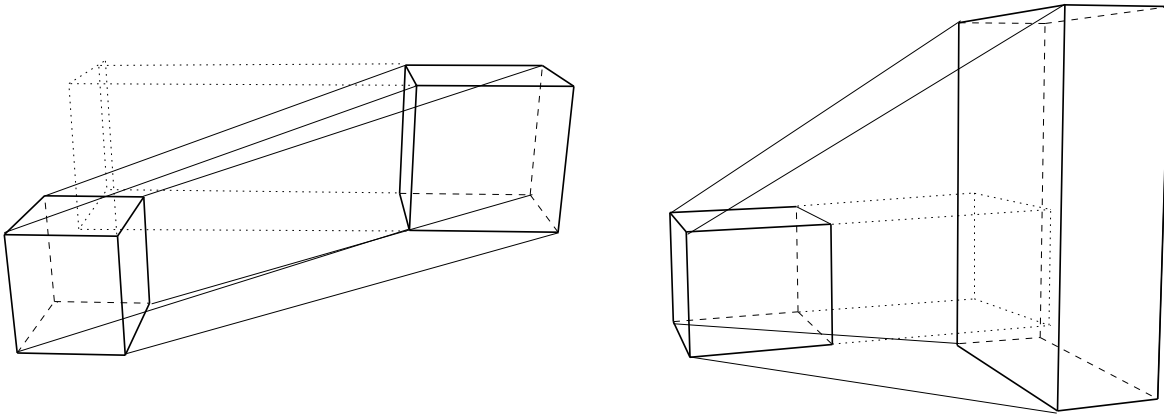


FIG. 2.8: Exemples de faisceaux joignant deux boîtes englobantes, respectivement constitués de six (à gauche) et quatre plans (à droite). Cette figure est extraite de la thèse de Soler [106].

où l'objet en mouvement est répliqué autant de fois que l'on souhaite calculer d'images. Une série de droites traversant la scène est ensuite choisie aléatoirement. Chacune d'elles est alors utilisée pour calculer un transfert d'énergie pour chaque image. La cohérence temporelle est ainsi assurée puisque l'échantillonnage aléatoire de la géométrie effectué reste le même d'une image l'autre.

Plus récemment, Myskowski *et al.* [78] proposent, afin de garantir la cohérence temporelle pour des animations utilisant une méthode de simulation stochastique de l'éclairage, l'emploi d'un filtrage temporel basé sur des critères perceptuels. Leur algorithme permet à la fois d'accélérer les calculs et de réduire les discontinuités temporelles pour des animations utilisant une méthode de tracé de photons. Toutefois, la cohérence temporelle n'est pas assurée de façon parfaite.

Méthode de radiosité

Curieusement, assez peu d'algorithmes exploitant la connaissance des trajectoires des objets ont été proposés pour le calcul de l'éclairage diffus par la méthode de radiosité depuis [10] en 1986. Dans ce dernier, Baum *et al.* proposent d'utiliser des mémoires tampon afin de limiter la quantité de calculs lors de la détermination des facteurs de forme par la méthode des hémicubes. Dans un premier temps, pour une surface donnée, on projette l'ensemble des surfaces fixes sur l'hémicube. La projection des volumes englobant les trajectoires des objets en déplacement permet ensuite de déterminer quels sont les facteurs de forme à recalculer.

Dans [84], Orti *et al.* utilisent le complexe de visibilité pour prédire les changements de visibilité dans une scène dynamique, mais leur étude se limite au cas bidimensionnel [60].

Nimeroff *et al.* [80] présentent une méthode calculant séparément l'éclairage direct et indirect. Ce dernier est calculé par une méthode de radiosité hiérarchique pour différentes dates espacées dans le temps. L'éclairage direct est ensuite calculé pour ces dates clés, de différents points de vus clés. Le rendu pour une date quelconque et d'un point de vue quelconque est ensuite fait par interpolation des images clés ainsi obtenues, d'une façon similaire à la méthode de Chen *et al.* [20]. Dans [81], Nimeroff *et al.* présentent des méthodes pour le choix des points de vus clés ainsi que pour celui des intervalles entre images clés en fonction des mouvements dans la scène.

2.3 Conclusion

Nous avons, dans ce chapitre, montré que les variations dans les échanges lumineux dans une animation présentent une cohérence importante. De ce fait, il doit être possible de bénéficier de la connaissance *a priori* du mouvement des objets pour construire un algorithme permettant d'effectuer rapidement et précisément le calcul de l'éclairage global diffus pour une scène animée. Une approche hiérarchique nous semble particulièrement appropriée puisque nous avons montré que pour une scène donnée, de nombreuses vitesses de variations différentes sont observées dans les échanges lumineux.

De plus, nous avons vu que relativement peu de travaux ont été consacrés jusqu'à présent au calcul d'une solution d'éclairage de qualité pour les animations par la méthode de radiosité. Les solutions existantes sont soit trop anciennes pour bénéficier des progrès les plus récents effectués dans le domaine, ou bien sont destinées à la mise à jour interactive de solutions existantes. Pour ces dernières, un temps de réponse rapide est privilégié par rapport à la qualité du résultat. De plus, elles ne prennent pas en compte la connaissance du scénario dont on peut bénéficier lorsque l'on calcule un film d'animation.

En conséquence, nous nous consacrerons, dans ce document, au développement d'un algorithme de calcul de l'éclairage global par une méthode inspirée de l'algorithme de radiosité hiérarchique. Pour ce faire, nous utiliserons la cohérence temporelle des échanges lumineux pour effectuer les calculs d'éclairage à un niveau de détail temporel nécessaire et suffisant. Un tel algorithme pourra, par exemple, permettre de fournir avec précision les termes d'éclairage ambiant à un algorithme de rendu de qualité basé sur un modèle local (tampon de profondeur, lancé de rayons, *etc.*), ou bien être utilisé directement pour la production d'une animation, si l'on emploie une méthode de reconstruction (récolte finale [103, 90, 72]) une fois la solution diffuse calculée. Puisqu'une des caractéristiques de l'algorithme de radiosité est de permettre le calcul d'une solution qui ne soit pas dépendante du point de vue, il pourra également permettre de calculer précisément les conditions d'éclairage ambiant dans le cas où l'on souhaite insérer des personnages réels dans une scène de synthèse animée.

Pour fournir un éclairage de qualité pour les animations, nous devons être particulièrement attentifs à ce que notre algorithme n'introduise pas de discontinuités gênantes dans la représentation temporelle de la radiosité. Ici aussi, la connaissance des trajectoires des éléments pourra certainement être mise à profit.

Deuxième partie

Radiosité hiérarchique spatiale et temporelle

Radiosité spatiale et temporelle : mise en équations

Nous étudions dans cette thèse le problème du calcul de l'illumination globale pour des animations dont le scénario (*i.e.* les trajectoires des objets ainsi que l'évolution des paramètres dans le temps) est connu par avance. Ce problème est généralement formulé comme une suite d'équations à résoudre, exprimant un équilibre instantané à chaque date correspondant à une image de l'animation. Dans ce chapitre, nous mettrons à profit la connaissance *a priori* des données temporelles, pour formuler notre problème comme une unique équation intégrale. Ainsi la solution calculée ne sera pas un échantillonnage temporel de la fonction d'illumination, mais bien une approximation continue¹ de cette fonction.

Dans ce chapitre nous présentons les notions mathématiques qui fondent les calculs de radiosité sur des intervalles de temps finis. Dans un premier temps, nous donnerons une nouvelle formulation de l'équation de la radiosité (1.8), étendue au cas des scènes animées. Puis, nous nous attacherons à la discrétisation de cette équation, suivant une méthode similaire à celle employée pour la méthode de radiosité et décrite dans la section 1.2.1. Finalement, nous étudierons plus particulièrement la discrétisation de cette équation dans une base de fonctions constantes par morceaux.

Même si les échanges radiatifs sont toujours considérés comme instantanés (c'est-à-dire que l'équilibre énergétique est à tout instant respecté) nous introduisons un couplage temporel entre les éléments radiatifs discrets, de façon à permettre la construction d'une hiérarchie mettant en oeuvre des éléments de durées différentes.

¹Le terme « *continu* » est ici compris comme l'opposé de *discret*

3.1 Illumination diffuse 4D : formulation continue

3.1.1 Hypothèses et notations

Nous voulons calculer la quantité de lumière sortante pour un ensemble de surfaces idéalement diffuses, à chaque instant dans un intervalle de temps donné. En d'autres termes nous voulons trouver une fonction $B(p, t)$ indiquant la radiosité au point p à l'instant t , définie sur l'ensemble $(\mathcal{M} \times \mathcal{T})$, où \mathcal{M} est l'ensemble de tous les points formant les surfaces de la scène, et $\mathcal{T} = [t^{initial}, t^{final}]$ est l'intervalle de l'animation durant lequel nous voulons effectuer le rendu de la scène.

De plus, puisque nous plaçons cette étude dans le cadre des animations dont le scénario nous est connu à l'avance, on supposera disposer des fonctions suivantes :

- $M : (\mathcal{M} \times \mathcal{T}) \rightarrow \mathbb{R}^3$ donnant la position du point p à l'instant t
- $\vec{N} : (\mathcal{M} \times \mathcal{T}) \rightarrow \mathcal{S}$ la normale sortante du point p à l'instant t
- $\rho : \mathcal{M} \rightarrow [0, 1]$ la valeur de la réflectance diffuse en chaque point
- $E : (\mathcal{M} \times \mathcal{T}) \rightarrow \mathbb{R}^+$ la radiosité émise au point p à l'instant t ne provenant pas d'une redistribution de l'énergie lumineuse incidente.

Dans les paragraphes suivants, nous utiliserons les définitions complémentaires suivantes (cf. figure 3.1) :

- $r(p, q, t)$ est la distance séparant p et q à l'instant t .

$$r : \begin{cases} (\mathcal{M} \times \mathcal{M} \times \mathcal{T}) & \rightarrow \mathbb{R}^+ \\ (p, q, t) & \rightarrow \|M(q, t) - M(p, t)\| \end{cases}$$

- $\vec{\omega}(p, q, t)$ est la direction de q vers p à l'instant t :

$$\vec{\omega} : \begin{cases} (\mathcal{M} \times \mathcal{M} \times \mathcal{T}) & \rightarrow \mathcal{S} \\ (p, q, t) & \rightarrow \frac{1}{r(p, q, t)}(M(q, t) - M(p, t)) \end{cases}$$

- $\theta(p, q, t)$ est l'angle entre les vecteurs $\vec{N}(p, t)$ et $\vec{\omega}(p, q, t)$

$$\theta : \begin{cases} (\mathcal{M} \times \mathcal{M} \times \mathcal{T}) & \rightarrow [0, \pi] \\ (p, q, t) & \rightarrow \arccos(\langle \vec{N}(p, t), \vec{\omega}(p, q, t) \rangle) \end{cases}$$

- $d(p, t, \vec{\omega})$ est la fonction indiquant la distance maximale pouvant être parcourue dans le vide à partir de p à l'instant t dans la direction $\vec{\omega}$

$$d : \begin{cases} (\mathcal{M} \times \mathcal{T} \times \mathcal{S}) & \rightarrow \mathbb{R}^+ \\ (p, t, \vec{\omega}) & \rightarrow \inf_{\alpha \in \mathbb{R}_*^+} \{ \alpha \in \mathbb{R}_*^+ ; (M(p, t) + \alpha \cdot \vec{\omega}) \in M(\mathcal{M}, t) \} \end{cases}$$

- $v(p, q, t)$ est la fonction indiquant la visibilité :

$$v : \begin{cases} (\mathcal{M} \times \mathcal{M} \times \mathcal{T}) & \rightarrow \{0, 1\} \\ (p, q, t) & \rightarrow 1 \text{ si } r(p, q, t) = d(p, \vec{\omega}(p, q, t)) \text{ et } \theta(p, q, t) > \frac{\pi}{2} \\ & \rightarrow 0 \text{ sinon} \end{cases}$$

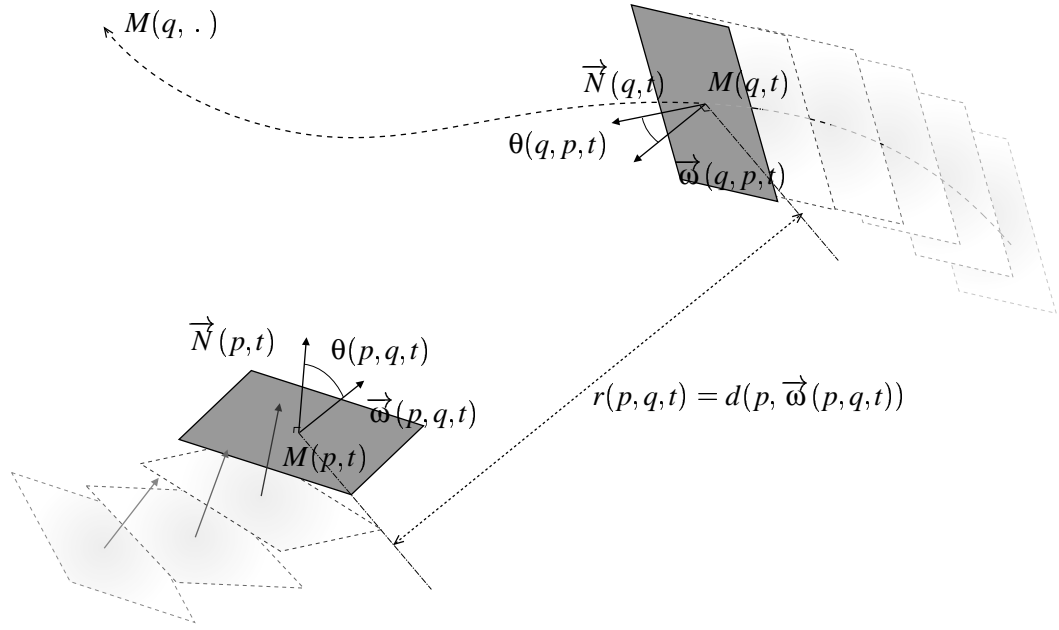


FIG. 3.1: Notations pour la description des animations

3.1.2 Mise en équation

Puisque l'on s'intéresse désormais au cas des animations, nous devons réécrire les équations données au paragraphe 1.2 afin d'introduire la variable temporelle. En particulier, la radiosit , ainsi que les positions des points et les fonctions qui en d coulent deviennent d sormais des fonctions du temps. Le temps de propagation de la lumi re entre deux objets  tant n gligeable devant celui s parant deux images de notre animation, on supposera que l' quilibre  nerg tique est r alis    tout instant. Dans une sc ne purement diffuse, si l'on n glige les effets dus   la phosphorescence (diffusion diff r e de la lumi re par les surfaces), l' quation de la radiosit  est donc v rifi e   tout instant t :

$$\forall t \in \mathcal{T}, \forall p \in \mathcal{M}, B(p,t) = E(p,t) + \rho(p) \int_{\mathcal{M}} B(q,t) k(p,q,t) dq$$

avec k le noyau d fini par :

$$k(p,q,t) = \frac{\cos \theta(p,q,t) \cos \theta(q,p,t)}{\pi r(p,q,t)^2} v(p,q,t) \quad (3.1)$$

Cette  quation est d'une forme diff rente de celle de la radiosit  dans le cas statique, puisque la sommation des contributions n'a lieu que sur le domaine spatial \mathcal{M} . Nous pouvons toutefois, gr ce aux notations d finies au paragraphe pr c dent, r crire cette  quation de la radiosit  sous la forme :

$$\forall X = (p,t) \in (\mathcal{M} \times \mathcal{T}) \quad B(X) = E(X) + \int_{Y=(q,t') \in (\mathcal{M} \times \mathcal{T})} B(Y) \mathcal{K}(X,Y) dY \quad (3.2)$$

o  :

- \mathcal{K} est une fonction d finie sur $(\mathcal{M} \times \mathcal{T})^2$ par

$$\mathcal{K}((p,t), (q,t')) = \rho(p) k(p,q,t) \delta(t,t') \quad (3.3)$$

- $\delta(t,t')$ est  gal   1 quand $t = t'$ et 0 autrement.

On remarquera que l'équation (3.2) semble décrire un transfert «inter-temporel» de lumière puisque t peut différer de t' . L'emploi de la fonction de Dirac $\delta(t, t')$ permet de réduire à zéro ces échanges. L'équation (3.2) exprime de ce fait l'équilibre énergétique indépendamment pour chaque instant.

De plus, on notera que cette équation est, de même que l'équation de la radiosité classique (1.8), une équation de Fredholm de seconde espèce et que l'opérateur K associé est de norme inférieure à 1. En effet, à chaque instant l'énergie retransmise est inférieure à l'énergie incidente (cf. paragraphe 1.1.2). Nous sommes donc assurés de l'existence et de l'unicité d'une solution à l'équation (3.2).

3.2 Discrétisation

Une approche similaire à celle que nous allons présenter ici (définition d'un maillage d'éléments sur l'espace spatial et temporel $(\mathcal{M} \times \mathcal{T})$) a été développée par Tsingos *et al.* [116, 117], pour une application différente : la simulation d'échanges énergétiques temporels dans le cadre de simulations acoustiques. Toutefois, dans ce cas particulier, il n'est pas possible de négliger le temps de propagation de l'énergie. De plus, dans ce cas la géométrie est fixe et on ne s'intéresse qu'aux variations énergétiques provoquées par un changement d'intensité des sources (auralisation d'un signal sonore dans un environnement 3D).

Nous allons discrétiser l'équation 3.2 suivant la méthode de Galerkin, de la même façon que pour l'algorithme de radiosité classique. Étant donné un produit vectoriel $\langle \cdot, \cdot \rangle$ pour l'espace vectoriel des fonctions sur $(\mathcal{M} \times \mathcal{T})$ et à valeurs dans \mathbb{R} , et un ensemble de fonctions $(u_i)_{1 \leq i \leq N}$ formant une famille orthogonale dans cet espace, nous voulons calculer l'approximation \tilde{B} dans $\mathcal{V}((u_i)_{1 \leq i \leq N})$ (l'espace vectoriel euclidien engendré par les u_i) de B telle que le résidu :

$$r(X) = \tilde{B}(X) - E(X) - \int_Y \tilde{B}(Y) \mathcal{K}(X, Y) dY \quad (3.4)$$

soit orthogonal à tous les u_i . Nous obtenons alors un système d'équations :

$$\forall i \in [1, N] \quad \langle r, u_i \rangle = \langle \tilde{B}, u_i \rangle - \langle E, u_i \rangle - \left\langle \int_Y \mathcal{K}(\cdot, Y) \tilde{B}(Y) dY, u_i \right\rangle$$

Or $\tilde{B} \in \mathcal{V}((u_i)_{1 \leq i \leq N})$ peut s'exprimer comme une combinaison linéaire des fonctions de base u_j :

$$\tilde{B} = \sum_{j=1}^N B_j u_j$$

donc

$$\forall i \in [1, N] \quad \sum_{j=1}^N (\langle u_i, u_j \rangle - \left\langle \int_Y \mathcal{K}(\cdot, Y) u_j(Y) dY, u_i \right\rangle) B_j = \langle E, u_i \rangle$$

Or les u_i forment une base orthogonale donc :

$$\forall (i, j) \in [1, N]^2, \quad \langle u_i, u_j \rangle = \|u_i\| \delta_{i,j}$$

L'équation de la radiosité spatio-temporelle discrète peut alors être exprimée comme une équation matricielle comparable à celle de la radiosité traditionnelle :

$$(I - M)B = E \quad (3.5)$$

avec

$$\forall i \in [1, N] \quad E_i = \frac{\langle E, u_i \rangle}{\|u_i\|^2}$$

et

$$\forall (i, j) \in [1, N]^2 \quad M_{i,j} = \frac{\langle \int_Y \mathcal{K}(\cdot, Y) u_j(Y) dY, u_i \rangle}{\|u_i\|^2} \quad (3.6)$$

3.3 Un exemple de discrétisation : Base d'éléments constants

3.3.1 Équations discrètes de la radiativité 4D

Nous allons maintenant exprimer l'équation de la radiativité (3.5) dans le cas d'une base de fonctions constantes. Dans ce cas, les fonctions u_i sont les fonctions indicatrices des éléments, c'est-à-dire les fonctions valant 1 sur l'élément considéré et 0 en dehors. Il s'agit bien d'une base orthogonale, car les éléments sont disjoints, mais non orthonormale, pour le produit scalaire canonique $\langle f, g \rangle = \int f(X)g(X)dX$.

De plus, nous supposons que notre scène est composée de surfaces effectuant un mouvement rigide. Nous pouvons alors décomposer l'espace $(\mathcal{M} \times \mathcal{T})$ en N éléments P_i sur lesquels la radiativité est supposée uniforme. Chaque élément P_i est défini par un intervalle temporel $T_i = [t_i^{initial}, t_i^{final}] \subset \mathcal{T}$ et par sa géométrie (éventuellement en mouvement) $Q_i \subset \mathcal{M}$. On remarquera que puisque le mouvement du support de Q_i est rigide, son aire est constante, et que tout élément est un produit cartésien de la forme $P_i = Q_i \times T_i$. Toute intégrale sur P_i peut donc être écrite sous la forme d'une double intégrale sur Q_i et T_i . Ainsi, on a :

$$\begin{aligned} \forall (i, j) \in [1, N]^2 \quad \langle u_i, u_j \rangle &= \int_{t_i^{initial}}^{t_i^{final}} \int_{Q_i} dx dt \delta_{i,j} \\ &= A_i(t_i^{final} - t_i^{initial}) \delta_{i,j} \end{aligned}$$

Supposons de plus que la fonction de réflectance ρ est uniforme sur chaque élément (en temps et en espace donc). Nous pouvons alors expliciter la valeur des coefficients de la matrice M donnée par l'équation (3.6).

$$\langle \int_{Y \in (\mathcal{M} \times \mathcal{T})} \mathcal{K}(\cdot, Y) u_j(Y) dY, u_i \rangle = \int_X \int_Y \mathcal{K}(\cdot, Y) u_j(Y) u_i(X) dY dX$$

nous donne, puisque u_i est nulle en dehors de $P_i = Q_i \times T_i$:

$$\langle \int_Y \mathcal{K}(\cdot, Y) u_j(Y) dY, u_i \rangle = \int_{T_i \cap T_j} \int_{Q_i} \int_{Q_j} \frac{\cos\theta(p, q, t) \cos\theta'(q, p, t)}{\pi r(p, q, t)^2} v(p, q, t) dq dp dt$$

et finalement on peut donner aux coefficients de M une forme semblable à celle donnée par l'équation 1.13 :

$$M_{i,j} = \rho_i \mathcal{F}_{i,j} \quad (3.7)$$

avec $\mathcal{F}_{i,j}$, le « facteur de forme spatio-temporel » entre P_i et P_j , valant :

$$\mathcal{F}_{i,j} = \frac{1}{A_i(t_i^{final} - t_i^{initial})} \int_{T_i \cap T_j} \int_{Q_i} \int_{Q_j} \frac{\cos\theta(p, q, t) \cos\theta'(q, p, t)}{\pi r(p, q, t)^2} v(p, q, t) dq dp dt \quad (3.8)$$

Intuitivement, on peut comprendre cette nouvelle formulation du facteur de forme comme une moyenne temporelle des facteurs de forme entre les deux surfaces en mouvement, pris à chaque instant à l'intervalle de temps commun aux deux éléments (de même que l'on pouvait voir le facteur de forme entre deux surfaces fixes comme une moyenne sur le récepteur de facteurs de forme d'une surface vers un point).

3.3.2 Propriétés du facteur de forme 4D

On remarquera que le « facteur de forme » entre deux éléments définis sur des intervalles de temps disjoints est nul. De plus, les $\mathcal{F}_{i,j}$ vérifient les mêmes propriétés algébriques que dans le cas statique.

1. La somme des $\mathcal{F}_{i,j}$ pour i fixé est inférieure ou égale à 1.
2. Quelle que soit la paire d'éléments (i, j) , on a la relation de couplage des facteurs de forme $A_i(t_i^{final} - t_i^{initial}) \mathcal{F}_{i,j} = A_j(t_j^{final} - t_j^{initial}) \mathcal{F}_{j,i}$.
3. Quelle que soit la paire d'éléments (i, j) , si on subdivise j en deux éléments j' et j'' , que cela soit par subdivision du support de j ou de son intervalle de définition, on a la relation d'addition des facteurs de forme : $\mathcal{F}_{i,j} = \mathcal{F}_{i,j'} + \mathcal{F}_{i,j''}$.

La borne supérieure sur la somme des facteurs de forme se déduit du résultat connu dans le cas statique. En effet :

$$\forall i \in [1, N], \bigcup_{j=1}^N (T_i \cap T_j) = T_i \cap \left(\bigcup_{j=1}^N T_j \right) \subseteq T_i$$

d'où

$$\forall i \in [1, N], \sum_{j=1}^N \mathcal{F}_{i,j} \leq \frac{1}{A_i(t_i^{final} - t_i^{initial})} \int_{t_i^{initial}}^{t_i^{final}} \int_{Q_i} \left(\sum_{j=1}^N \int_{Q_j} \frac{\cos\theta \cos\theta'}{\pi r(p, q, t)^2} v(p, q, t) dq \right) dp dt \quad (3.9)$$

Or on sait qu'à tout instant t , la somme des facteurs de forme de toutes les surfaces de la scène \mathcal{M} vers un polygone quelconque Q_i est inférieure ou égale à 1 (cf. équations 1.15). Donc on a :

$$\forall t \in [t_i^{initial}, t_i^{final}], \int_{Q_i} \left(\sum_{j=1}^N \int_{Q_j} \frac{\cos\theta \cos\theta'}{\pi r(p, q, t)^2} v(p, q, t) dq \right) dp \leq A_i \quad (3.10)$$

Des inégalités 3.9 et 3.10 on obtient :

$$\forall i \in [1, N], \sum_{j=1}^N \mathcal{F}_{i,j} \leq \frac{1}{A_i(t_i^{final} - t_i^{initial})} \int_{t_i^{initial}}^{t_i^{final}} A_i dt = 1 \quad \square$$

Ceci nous garantit que la matrice $(I - M)$ est à diagonale dominante, donc inversible. Le système linéaire auquel nous a mené le processus de discrétisation possède donc une unique solution.

On notera que, de même que dans le cas statique, la borne supérieure est atteinte lorsque la configuration des surfaces est à tout instant telle que, en présence de réflecteurs diffus parfaits, la quantité d'énergie réémise dans la scène sera égale à la quantité d'énergie incidente, c'est-à-dire lorsque les surfaces forment à tout instant un *système clos*.

De même la relation de couplage des facteurs de forme se déduit aisément de la propriété correspondante dans le cas statique. La propriété d'additivité quant à elle est une conséquence de l'additivité des intégrales sur supports disjoints.

3.4 Discussion

Nous avons dans ce chapitre reformulé l'équation de radiosit  sur un intervalle de temps fini. Cette nouvelle formulation permet de consid rer les  changes lumineux au cours d'une animation dans leur ensemble, afin de permettre la recherche d'une solution d finie sur un espace temporel continu, plut t qu'un  chantillonnage de cette m me solution   chaque image de l'animation. Il nous sera ainsi par la suite plus ais  de profiter de la coh rence temporelle de l' clairage, ainsi que de s'assurer de la continuit  des solutions calcul es. Nous avons pour cela d fini un nouvel op rateur int gral exprimant la redistribution de l' nergie lumineuse re ue par l'ensemble des surfaces de la sc ne au cours de l'int gralit  de la dur e de l'animation. Nous avons ensuite, en vue d'une r solution num rique du probl me ainsi pos , discr tis  cette  quation suivant la m thode de Galerkin, d'une fa on similaire   celle employ e pour le cas statique.

  la d finition du facteur de forme pr s, nous avons ainsi ramen  le probl me du calcul d'une approximation de la fonction de radiosit  pour une animation *formellement* aux m mes calculs que ceux n cessaires   la g n ration d'une solution pour une sc ne statique. Toutefois, la nature diff rente du maillage, et le fait que sa taille soit singuli rement plus importante (du fait de l'augmentation de la dimensionnalit  du probl me) fait que sa r solution est *effectivement* beaucoup plus complexe que dans le cas statique.

M me si la transposition na ve d'un algorithme permettant de r soudre le syst me d' quations 1.13 dans le cas quadridimensionnel devrait le plus souvent  tre possible (  l'exception notable des m thodes impliquant un maillage de discontinuit ²), il est de toute  vidence pr f rable de se servir des particularit s du cas quadridimensionnel pour construire un algorithme de r solution efficace :

- Bien que de taille plus importante, la matrice des coefficients de couplage dans le cas de la radiosit  spatiale et temporelle est beaucoup plus creuse que dans le cas classique. En effet, pour deux  l ments i et j dont les intervalles de temps sont disjoints, les coefficients d'interaction $M_{i,j}$ et $M_{j,i}$ sont nuls. Ainsi, les m thodes permettant de ne repr senter en m moire que les interactions dont on sait qu'elles correspondent   un facteur de forme strictement positif seront probablement les seules envisageables en pratique.
- Contrairement   ce qui se passe dans les dimensions spatiales, il est possible que la fonction de radiosit  B cherch e soit en certains points effectivement constante en t (par exemple dans le cas ou  metteur et r cepteur sont fixes dans un m me r f rentiel). Il est donc possible que le raffinement du maillage dans la dimension temporelle n'apporte pas toujours un gain de pr cision. En revanche, un maillage plus fin implique toujours une augmentation de la m moire n cessaire et du temps de calcul. Un traitement s par  des dimensions spatiales et temporelles du probl me est donc n cessaire si l'on souhaite maximiser l'efficacit  de l'algorithme.

Nous allons dans les chapitres suivants d velopper une m thode hi rarchique de r solution de ce syst me d' quations, inspir es des m thodes d velopp es pour le cas statique et pr sent es au paragraphe 1.2. Nous nous attacherons dans un premier temps   la description de l'adaptation de l'algorithme pr sent  par Hanrahan *et al.* [55]. Apr s une analyse des performances et des limitations de cet algorithme, et afin d'y rem dier, nous montrerons comment nous avons adapt  au cas des animations certaines des extensions de l'algorithme de radiosit  hi rarchique.

²Pour un tel maillage en dimension 4, les dimensions spatiale et temporelle ne seront plus s parables : les zones de discontinuit s de la fonction de visibilit  dues   des objets en mouvement, sur lesquelles doivent s'appuyer les fronti res des  l ments de maillage n' tant pas orthogonales   \mathcal{M} ou   \mathcal{T} .

Algorithme de résolution hiérarchique spatial et temporel

Nous allons dans ce chapitre proposer une adaptation de l'algorithme de radiosit  hi rarchique de Hanrahan *et al.* [55] au cas de la radiosit  temporelle. On peut raisonnablement attendre d'une telle transposition l'obtention d'un algorithme efficace pour r soudre l' quation 3.5. En effet, comme nous l'avons soulign  au paragraphe 3.4, le probl me de la radiosit  spatiale et temporelle tel que nous l'avons formul  est *formellement* de m me nature que celui de la radiosit  dans le cas statique. De plus, nous avons montr  au paragraphe 2.1.1 que la coh rence temporelle des  changes lumineux dans une sc ne anim e est non seulement suffisante pour que l'on puisse factoriser une partie importante des calculs effectu s, mais aussi d'une nature suffisamment complexe pour qu'une repr sentation multi-r solution des variations des  changes lumineux au cours du temps soit justifi e.

Afin de construire un tel algorithme il nous faut d finir :

- une structure de donn es ad quate pour m moriser et acc der aux valeurs de la radiosit  sur des  l ments spatio-temporels   diff rents niveaux de pr cision tant en temps qu'en espace. Cette structure doit permettre un raffinement adaptatif aussi bien dans la dimension spatiale que temporelle.
- une structure de liens repr sentant les  changes lumineux entre  l ments de la sc ne, permettant leur positionnement au juste niveau dans cette hi rarchie et le stockage de leur valeur. Nous devons  galement d finir pour cette structure une politique de raffinement adapt e aux sp cificit s de la formulation 4D.

Nous d veloppons dans ce chapitre les probl mes soulev s par ces deux points, et pr senterons les solutions algorithmiques que nous avons appliqu es dans notre impl mentation.

4.1 Hiérarchie des éléments

Nous nous attacherons, dans ce paragraphe, à la définition d'une structure de données permettant la représentation multi-résolutions de la fonction de radiosité discrète que nous avons définie au paragraphe 3.2. La discrétisation de l'équation de radiosité, telle que nous l'avons effectuée, repose sur la construction d'un maillage de l'espace ($\mathcal{M} \times \mathcal{T}$). Nous devons faire en sorte de permettre la construction et la manipulation de ce maillage à différents niveaux de précision. La solution employée dans [55] consiste à définir un mécanisme de subdivision de chaque élément de surface en plusieurs éléments compacts plus fins dont l'union couvre le même domaine et dont l'intersection est de mesure nulle. Ainsi, par application répétée du processus de subdivision, on obtient une succession de maillages de précision de plus en plus fine, pour laquelle il est aisé de passer d'un niveau de représentation à l'autre (dans le cas de fonctions de bases constantes par morceaux, il s'agit juste de moyenner les valeurs de la fonction sur les éléments fils pour obtenir celle sur l'élément père, cf. paragraphe 1.2.3).

Nous montrerons dans un premier temps que l'on peut employer une structure similaire pour permettre une représentation multi-résolution de notre maillage spatial et temporel. Nous préciserons le mécanisme de subdivision employé pour la production des différents niveaux de précision. Nous indiquerons également comment la cohérence entre les différents niveaux de représentation est assurée. Nous terminerons en montrant que cette structure de donnée peut être aisément et rapidement traversée pour permettre l'affichage de la solution à une date donnée.

4.1.1 Choix d'une structure

Rappelons que (cf. paragraphe 3.3) les éléments de notre maillage sont définis par leur description géométrique ainsi que par leur étendue temporelle (*i.e.* l'intervalle de temps pendant lequel leur radiosité est supposée constante). Ils peuvent de ce fait être subdivisés soit dans l'espace (par exemple, en quatre sous-éléments dans le cas de quadrilatères ou de triangles), soit dans le temps (en divisant leur étendue temporelle et en laissant leur géométrie inchangée).

Il est ainsi aisé de concevoir une relation hiérarchique entre éléments de notre maillage sur ($\mathcal{M} \times \mathcal{T}$). Un élément subdivisé sera le « père » des éléments créés directement par cette subdivision. Toutefois, il nous faut décider comment concilier la relation de paternité due aux subdivisions spatiales et celle due aux subdivisions temporelles.

Il est possible d'envisager de subordonner une des deux hiérarchies à l'autre. Ainsi, Finkelstein *et al.* [32] proposent, pour une application différente, de créer une hiérarchie temporelle de hiérarchies spatiales. Une telle subordination se justifie lorsque l'on peut procéder à un important partage de données entre éléments (le mécanisme de compression destructive permet à Finkelstein *et al.* de créer un nombre important de « liens transversaux » entre les hiérarchies temporelles, leur permettant d'éviter des redondances). Toutefois, dans le cas de la radiosité spatiale et temporelle, il n'y a pas de raison manifeste pour privilégier une dimension plutôt qu'une autre. C'est pourquoi nous avons choisi d'« entrelacer » les hiérarchies spatiales et temporelles.

4.1.2 Mécanisme de subdivision

Un élément de notre hiérarchie peut avoir des descendants indifféremment dans le temps ou dans l'espace. La figure 4.1 montre un exemple simple de hiérarchie spatiale et temporelle. Un élément de maillage formé par le produit cartésien d'un quadrilatère et d'un intervalle de temps donné, est d'abord subdivisé temporellement. L'élément fils correspondant à la seconde moitié de l'intervalle de temps est alors subdivisé spatialement.

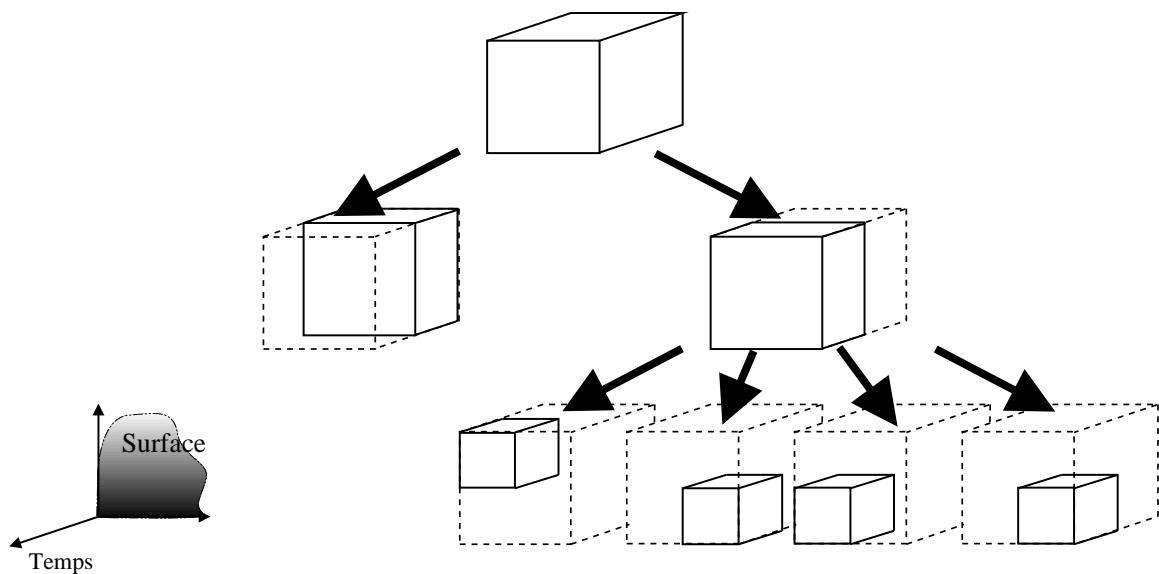


FIG. 4.1: Exemple de hiérarchie spatiale et temporelle

Nous avons implémenté ce mécanisme dans un simulateur de radiosité hiérarchique, utilisant une subdivision de type arbre quaternaire pour la géométrie et une subdivision en deux parties égales pour la description temporelle.

En ce qui concerne la place mémoire, on notera qu'il n'est nécessaire de dupliquer ni la représentation de la fonction M (indiquant la trajectoire de chaque point) ni les données indiquant les limites du domaine Q_i (par exemple les coordonnées des trois sommets sur le support dans le cas d'un élément triangulaire) à chaque subdivision temporelle. Les seules données qu'il soit nécessaire de créer lorsque l'on subdivise un élément dans le temps sont la description temporelle de ses deux fils et leurs radiosités. Toutes les données géométriques peuvent en effet être partagées entre les descendants et leur père.

Des méthodes de subdivision plus élaborées peuvent aisément être essayées, en particulier si l'on dispose d'informations concernant un événement particulier devant avoir lieu durant l'intervalle de temps considéré et sur lequel on souhaiterait aligner la séparation.

4.1.3 Maintenance de la cohérence hiérarchique

Muni d'une telle structure hiérarchique, l'algorithme pour résoudre l'équation 3.5 est le même que pour la radiosité hiérarchique classique. Des trois parties composant à l'origine une itération - c'est-à-dire le raffinement, la phase de collecte des irradiances (« *gathering* ») et la mise-à-jour de la hiérarchie (« *push-pull* ») - seule la procédure de raffinement des liens nécessite des changements spécifiques qui seront décrits au paragraphe 4.2.2.

Ainsi la mise à jour de la radiosité aux différents niveaux de la hiérarchie se déroule de façon fort similaire à celle décrite dans le paragraphe 1.2.3 : les irradiances cumulées reçues par un objet père sont transmises aux objets fils, et leurs radiosités sont en retour moyennées (pondérées par leur mesure, c'est-à-dire le produit de leur aire et de leur étendue temporelle) pour calculer la radiosité du père (cf. pseudo-code 4.1).

Algorithme 4.1 Mise à jour récursive de la hiérarchie

```

Mise_A_Jour(Donnees : element, somme_irradiance_peres, Resultat : energie_totale)
  energie_totale  $\leftarrow$  0
  element.irradiance_recoltee  $\leftarrow$  element.irradiance_recoltee + somme_irradiance_peres
  si ( Est_une_feuille(element) ) alors
    element.radiosite  $\leftarrow$  element.irradiance_recoltee  $\times$  element.reflectance
    energie_totale  $\leftarrow$  element.radiosite  $\times$  element.aire  $\times$  element.duree
  sinon
    pour tout f  $\in$  fils(element) faire
      Mise_A_Jour(f, element.irradiance_recoltee, energie_fils)
      energie_totale  $\leftarrow$  energie_totale + energie_fils
    fin pour
    element.radiosite  $\leftarrow$  energie_totale / (element.aire  $\times$  element.duree)
  fin si

```

4.1.4 Affichage de la solution

Une fois la solution calculée, l’affichage du résultat pour toute date t se fait simplement et rapidement : il suffit de parcourir la hiérarchie à la recherche des feuilles dont l’intervalle de temps contient t et de les afficher à la position donnée par la fonction M avec la couleur correspondant à leur radiosité. On notera que l’image obtenue est une visualisation d’une « tranche » de la solution calculée correspondante à un instant donné. Faire parcourir à la date d’affichage toute la durée de l’animation fait se modifier en conséquence la partie affichée du maillage, donnant l’illusion d’un maillage dépendant du temps (cf. figure 4.2).

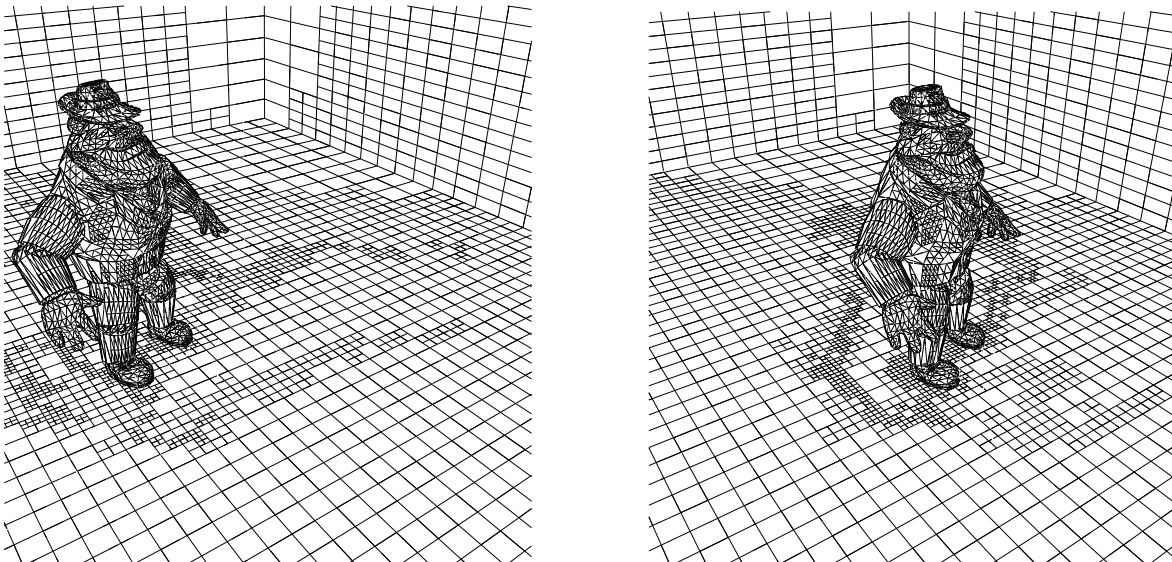


FIG. 4.2: Visualisation du maillage spatial et temporel à deux dates différentes

4.2 Échanges énergétiques

4.2.1 Structure de la hiérarchie des liens

De façon similaire au cas statique, les échanges énergétiques entre surfaces seront représentés par des liens reliant deux éléments. Leur structure restera la même que pour l'algorithme de radiosité hiérarchique classique (cf. paragraphe 1.2.3) : seuls nécessitent d'être mémorisés :

- une référence à l'élément source.
- le facteur de forme correspondant à l'interaction représentée.

Il faut toutefois noter qu'un lien représente désormais un échange énergétique entre deux éléments de notre maillage quadridimensionnel. Or, comme nous l'avons rappelé dans le paragraphe 3.1 l'équilibre énergétique doit être respecté à chaque instant. Il ne saurait y avoir transfert énergétique « intertemporel ». De ce fait, chaque lien représente un échange énergétique entre deux portions de surface (*i.e.* une partie de \mathcal{M}^2) durant un unique intervalle de temps fini (un segment $[t^{initial}, t^{final}] \in \mathcal{T}$, et non un produit cartésien de deux segments). Cet intervalle de temps, auquel nous ferons désormais référence comme étant l'*intervalle de validité* du lien, est exprimé par l'équation suivante (cf. équation (3.8) définissant le facteur de forme entre deux éléments) :

$$[t_{lien}^{initial}, t_{lien}^{final}] = [t_{émetteur}^{initial}, t_{émetteur}^{final}] \cap [t_{récepteur}^{initial}, t_{récepteur}^{final}] \quad (4.1)$$

4.2.2 Raffinement des liens

Un double problème

Pour notre algorithme comme pour celui de radiosité hiérarchique classique, le processus de raffinement est ici la partie la plus gourmande en temps de calcul. De son bon fonctionnement découle directement la qualité de la solution calculée. Un raffinement mal effectué produira une hiérarchie inutilement lourde et totalement inefficace.

De plus une approche spécifique au cas des animations doit être développée. Il nous est nécessaire :

1. de déterminer si un lien donné est une représentation suffisamment fidèle des échanges lumineux entre deux éléments de la hiérarchie,
2. de décider, dans le cas contraire qui, de l'étendue spatiale ou temporelle des dits éléments, est le plus « responsable » de l'inadéquation de cette représentation, et de choisir en conséquence entre une subdivision spatiale et une subdivision temporelle du lien.

Ces décisions ne sont en général pas faciles à prendre, et peuvent avoir un impact considérable sur la consommation mémoire ainsi que sur le temps de calcul nécessaire à l'obtention d'une solution.

Notre oracle de raffinement

Nous avons choisi un critère de raffinement basé sur une estimation de la variance des échanges radiatifs sur un élément dans les dimensions spatiales et temporelles. Nous échantillonnons la fonction géométrique k et la fonction de visibilité v (cf. paragraphe 3.1), ainsi que la radiosité de l'émetteur si ce dernier n'est pas une feuille de la hiérarchie (ceci nous permet de prendre en compte les variations d'éclairement indirect).

Il peut s'avérer problématique de se baser simplement sur l'échantillonnage : des bordures d'ombres peuvent être incorrectement détectées, provoquant des clignotements malvenus, ou encore les ombres en déplacement peuvent laisser des traces sur leur passage. En revanche, on peut raisonnablement s'attendre à ce que la visibilité ne dépende pas du temps pour un assez grand nombre de liens (par

exemple pour des surfaces très proches ou suivant le même déplacement, entre lesquelles ne passerait aucun obstacle), situation qu'il convient de détecter le plus haut possible dans la hiérarchie afin de s'épargner au maximum un coûteux échantillonnage temporel.

En conséquence, nous avons décidé d'employer des faisceaux (*shafts*) [53] afin de détecter sans erreurs les surfaces en mouvement entre les deux extrémités d'un lien donné, et de forcer le cas échéant une subdivision appropriée.

1. Si la source et le récepteur du lien considéré suivent le même mouvement, nous construisons le faisceau joignant leurs boîtes englobantes. Puis nous parcourons de façon hiérarchique l'ensemble des objets ne suivant pas la même trajectoire afin de déterminer s'ils rencontrent le faisceau et causent une occultation « significative », pendant une portion limitée de l'intervalle de temps correspondant à la durée du lien. Si un tel obstacle est trouvé alors nous forçons un raffinement temporel de ce lien.
2. Dans tous les autres cas, nous échantillons le produit de la fonction noyau k , la radiosité de l'émetteur et la fonction de visibilité v , pour un ensemble de dates t et de couples de points (p, q) . Puis nous calculons pour notre ensemble d'échantillons l'écart-type pour t fixé et l'écart-type pour (p, q) fixé. Nous comparons ces deux valeurs à un seuil pour décider si nous devons raffiner ce lien (nous raffinons donc lorsque les variations spatiales ou temporelles de l'énergie lumineuse échangée entre deux éléments sont importantes), puis l'une avec l'autre pour décider quel type de raffinement nous devons appliquer (nous cherchons, par une telle subdivision, à réduire au maximum ces variations).

Détails d'implémentation

Quelques optimisations sont nécessaires afin de réduire le nombre d'échantillons calculés :

- Si la source et le récepteur suivent le même déplacement (*cf.* figure 4.3), la fonction k ne dépend plus du temps (puisque nous ne considérons que des mouvements rigides, les termes angulaires et les distances ne varient pas au cours de ce mouvement).
- Si nous avons pu établir que notre faisceau n'est jamais traversé par un objet en mouvement, alors nous savons que la visibilité ne dépend pas du temps.

A moins qu'il n'y ait de nombreux grands objets en mouvement dans la scène, nous pouvons raisonnablement espérer qu'un grand nombre de liens bénéficieront de ces optimisations et donc que le processus de raffinement soit, pour eux, aussi rapide que pour le cas statique.

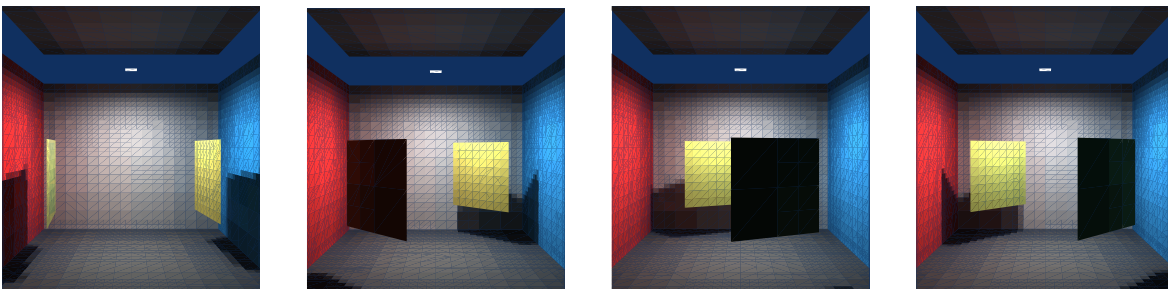


FIG. 4.3: Exemple de configuration permettant une optimisation : puisque les deux carrés jaunes tournent autour du même axe, les termes dus à la configuration géométrique (angles, distances) ne varient pas au cours du temps. De plus, aucun objet ne passera entre les deux. La visibilité restera donc constante. Ainsi, l'évaluation de l'oracle de raffinement pour cette interaction ne prendra pas plus de temps que si on l'effectuait pour une seule image

4.2.3 Détermination de la visibilité

L'évaluation de la fonction de visibilité v représente l'essentiel du temps de calcul de l'algorithme de radiosité hiérarchique. Il est donc essentiel d'utiliser une structure permettant l'accélération du lancer de rayon (cf. paragraphe 1.2.4). Celles traditionnellement employés pour des scènes statiques [36] ne peuvent pas toujours l'être en présence d'objets en mouvement. Les structures fixes telles que les grilles de voxels ou les *kd-arbres* peuvent, dans le cas d'un calcul incrémental être mises à jour pour tenir compte des déplacements des objets à chaque étape de l'animation de façon efficace [91], mais comme les requêtes de visibilité dans le cas de notre algorithme ne sont pas ordonnées temporellement, une telle mise à jour devrait avoir lieu à chaque lancer de rayon, ce qui s'avérerait beaucoup trop long.

En revanche, il est possible dans notre cas, de construire dans une phase préliminaire une hiérarchie de boîtes englobantes et de l'utiliser comme structure d'accélération sans avoir à y apporter de modifications ultérieures. En effet, nous avons fait l'hypothèse au paragraphe 3.3.1 que nos objets n'effectuent que des mouvements rigides. Si nous ne regroupons pas deux objets suivant des trajectoires différentes dans une même boîte englobante, une boîte englobante d'un objet à l'instant t est l'image de la même boîte englobante construite à l'instant initial, par la transformation qui amène l'objet considéré à sa position à l'instant t .

De plus, les mouvements rigides sont, dans le cas général, des compositions de translation et de rotation. La fonction M étant une union de telles transformations sur des supports disjoints, elle est donc aisément inversible (il s'agit juste d'effectuer un changement de signe du paramètre temporel lors de son évaluation). Plutôt que de calculer l'image d'une boîte englobante (transformation appliquée à trois points), nous calculerons donc plutôt l'image inverse du rayon (transformation appliquée à deux points). L'algorithme 4.2 permet d'échantillonner la fonction v entre deux points à un instant donné.

Algorithme 4.2 Évaluation de la visibilité dans une hiérarchie de boîtes englobantes en mouvement

Est_Visible(Données : p, q, t) → **booleen**

```

Resultat ← vrai
x ← M(p, t)
y ← M(q, t)
pour toute trajectoire  $M_i$  composant  $M$  faire
    | Boite_obstacle ← Boite_Englobante(Surfaces_Affectees_Par( $M_i$ ))
    | Resultat ← Resultat et non Oclusion( $M_i^{-1}(x,t), M_i^{-1}(y,t), Boite\_obstacle$ )
fin pour

```

Oclusion(Données : $x, y, Boite_mere$) → **booleen**

```

Resultat ← faux
pour toute Surface ∈ Contenu(Boite_mere) faire
    | Resultat ← Resultat ou Intersection( $x, y, Surface$ )
fin pour
pour toute Boite_fille ∈ Filles(Boite_mere) faire
    | si ( Intersection( $x, y, Boite\_fille$ ) ) alors
        | Resultat ← Resultat ou Oclusion( $x, y, Boite\_fille$ )
        fin si
fin pour

```

Pour la construction de la hiérarchie de boîtes englobantes, il est possible d'utiliser n'importe quel algorithme habituellement employé dans le cas statique [56]. Il faudra toutefois procéder séparément pour chaque ensemble d'objets fixes dans un même référentiel. Nous avons pour notre part, choisi d'employer une méthode inspirée de celle de Christensen [21], modifiée afin d'éviter la construction de boîtes ne contenant qu'une seule surface (*cf.* algorithme 4.3 et figure 4.4). Cette méthode procède en regroupant les surfaces en fonction de l'octant de la boîte originelle dans laquelle leurs centres de gravité se trouvent. Les surfaces trop grandes pour être insérées dans un octant (comparaison effectuée avec une tolérance ε paramétrable) seront placées à ce niveau de la hiérarchie. On construit ensuite les boîtes englobantes des huit listes obtenues, sur lesquelles on applique récursivement le processus.

Algorithme 4.3 Construction d'une hiérarchie de boîtes englobantes (d'après [21])

Construction(Données : *liste_surfaces*, *conteneur*)

```

Variables :
    | tableau_listes[8]
    | boite_fille
    | pour toute surface  $\in$  liste_surfaces faire
    |     | si ( Taille(surface)  $\geq$  Taille(conteneur)  $\times$  (1 +  $\varepsilon$ )/2 ) alors
    |     |     | Inclure(surface, conteneur)
    |     |     | sinon
    |     |     |     | Ajouter(surface, tableau_listes [ Octant(surface) ] )
    |     |     |     | fin si
    |     | fin pour
    | pour tout octant  $\in$  {1, ..., 8} faire
    |     | si ( Longueur(tableau_listes[octant] > 1 ) ) alors
    |     |     | boite_fille  $\leftarrow$  Boite_Englobante(tableau_listes[octant])
    |     |     |     | Construction(tableau_listes[octant], boite_fille )
    |     |     |     | Inclure(boite_fille, conteneur)
    |     |     | sinon
    |     |     |     | Inclure(Unique_Surface(tableau_listes[octant]), conteneur )
    |     |     | fin si
    | fin pour

```

4.3 Discussion

Nous avons dans ce chapitre présenté un nouvel algorithme permettant la résolution efficace du problème du calcul de la radiosité pour toutes les surfaces d'une scène animée tel que nous l'avons formalisé au chapitre 3. Nous montrerons au chapitre suivant dans quelle mesure cet algorithme permet l'accélération du calcul de solutions de radiosité pour une animation, par rapport à un calcul image par image avec un raffinement d'une précision équivalente. Nous y présenterons également quelques cas où son comportement n'est en pratique pas satisfaisant, et nous en analyserons les causes.

L'algorithme présenté dans ce chapitre est une généralisation de l'algorithme de radiosité de Hanrahan *et al.* [55] au cas des animations. À ce titre, son implémentation ne devrait pas présenter de difficulté à quiconque est familier avec l'algorithme original. En particulier, l'optimisation en est assez similaire. Les fonctions permettant l'évaluation de la visibilité représentent la majeure partie de son

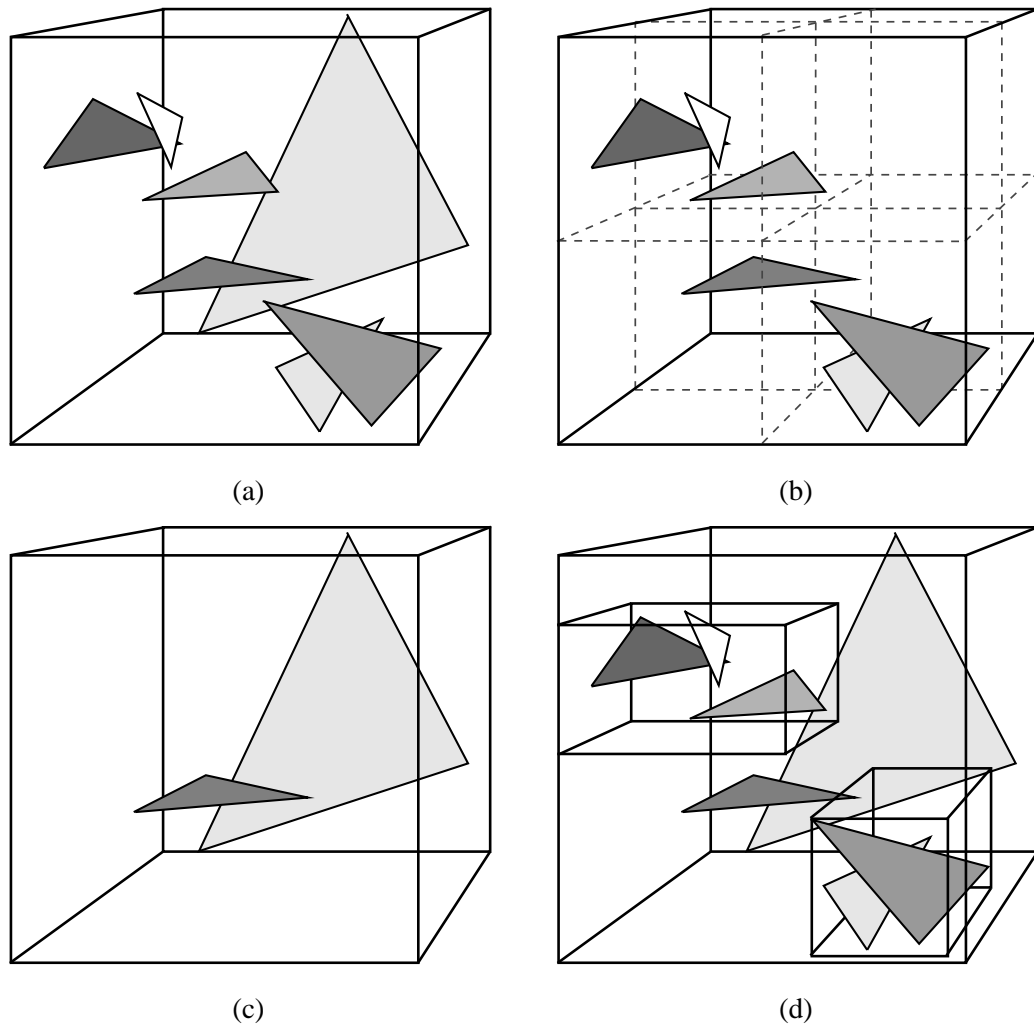


FIG. 4.4: Illustration de l’algorithme 4.3 : **(a)** Données, **(b)** Regroupement des petites surfaces par octant, **(c)** Les grandes surfaces et celles isolées dans leur octant restent à ce niveau de la hiérarchie, **(d)** Construction des boîtes englobantes avant la récursion.

coût. Le programmeur prêtera donc une attention particulière à leur mise au point. De même, l’oracle de raffinement joue ici aussi un rôle central. Les performances de l’algorithme seront fortement affectées par la construction d’une hiérarchie inadéquate due à un oracle trop de mauvais conseil. Le choix des paramètres (nombre d’échantillons de visibilité, limites du raffinement, *etc.*) devra être effectué avec discernement.

En revanche, il existe des points spécifiques de notre algorithme auxquels il faut également porter attention. Une amélioration possible provient du fait que notre hiérarchie traite à égalité subdivisions spatiales et temporelles. En effet, il est possible que le processus de raffinement recommande de subdiviser un élément en temps (respectivement en espace) alors qu’il a déjà été subdivisé en espace (respectivement en temps) à cause d’une interaction avec un autre objet. Pour le moment nous avons choisi de transmettre le lien à raffiner aux enfants existants et de reprendre le processus de subdivision à ce niveau, et ce récursivement jusqu’à ce qu’une feuille soit atteinte. Pour y remédier, il est possible d’imaginer l’ajout d’une passe de maintenance ayant lieu à la fin du raffinement et qui serait destinée à réordonner correctement la hiérarchie de façon à limiter le nombre de telles collisions.

La similitude formelle de notre algorithme avec l'algorithme de radiosité hiérarchique classique n'a pas pour seule conséquence de faciliter la tâche du programmeur. En effet, comme nous l'avons vu au début de ce mémoire (*cf.* paragraphe 1.2), une littérature abondante est consacrée à l'étude de l'algorithme de radiosité hiérarchique. La communauté scientifique a proposé de nombreuses variantes afin de palier à ses insuffisances. Il est donc possible, dans la plupart des cas, de transposer ces améliorations afin de les appliquer à notre algorithme. Nous en présenterons quelques exemples dans la dernière partie de ce mémoire.

Résultats et interprétation

DANS ce chapitre, nous montrerons que l’algorithme de radiosité hiérarchique spatiale et temporelle permet le calcul de l’éclairage global diffus pour les animations, plus rapidement que ne le permettrait un calcul image par image. Nous étudierons également les limitations de cet algorithme afin de permettre, dans les chapitres suivants, d’en améliorer les performances.

Pour ce faire, nous nous baserons sur des résultats expérimentaux obtenus à partir de l’exécution de notre algorithme sur différentes scènes de test. Ces dernières seront construites :

- soit pour être représentatives d’une certaine catégorie de scènes afin de prouver le bon fonctionnement de notre algorithme,
- soit pour mettre délibérément en évidence un point particulier nécessitant amélioration.

Nous commenterons ces résultats de manière à mettre en avant les particularités de notre algorithme. Les points essentiels dégagés lors de cette discussion nous serviront de base pour les développements des chapitres suivants. Par delà ces résultats numériques, il est important de souligner le fait que notre algorithme fournit le cadre pour l’expérimentation de méthodes plus élaborées qui nous permettront de bénéficier de la connaissance globale des variations d’illumination au cours du temps.

5.1 Efficacité de la méthode

5.1.1 Expérimentations

Nous avons appliqué notre algorithme aux deux scènes de test présentées au paragraphe 2.1.1 :

- La première (ACTEURS), a été construite afin d’étudier les variations d’éclairage dues au mouvement de deux acteurs (auxquels nous avons substitué deux boîtes de taille équivalente) dans une pièce.
- La seconde (ÉCRAN), comporte un panneau coulissant devant une source de lumière. Elle nous permet d’étudier le comportement de notre algorithme dans le cas où le mouvement d’une ombre projetée provoque de grands changements dans l’éclairage direct de la scène.

Ces deux animations durent 3 secondes chacune et représentent un total de 75 images. La figure 5.1 en montre quelques images représentatives.

5.1.2 Résultats

Nous avons comparé les résultats à ceux obtenus par un algorithme de radiosité hiérarchique classique générant séparément chaque image et utilisant les mêmes paramètres de raffinement. Les résultats, obtenus sur un processeur MIPS R10000 cadencé à 250Mhz, sont donnés dans le tableau suivant :

	Temps de calcul		Mémoire utilisée	
	Notre algorithme	Radiosité «statique»	Notre algorithme	Radiosité «statique»
ACTEURS	335 s	$11,5 \text{ s} \times 75 = 862,5 \text{ s}$	112 Mo	11 Mo
ÉCRAN	671 s	$18,1 \text{ s} \times 75 = 1357,5 \text{ s}$	237 Mo	14,3 Mo

TAB. 5.1: Performances de l'algorithme de radiosité 4D hiérarchique

En termes de qualité, la solution obtenue par la méthode de radiosité temporelle est similaire à celle obtenue image par image.

5.1.3 Commentaires

Ces résultats appellent les commentaires suivants :

Coût mémoire

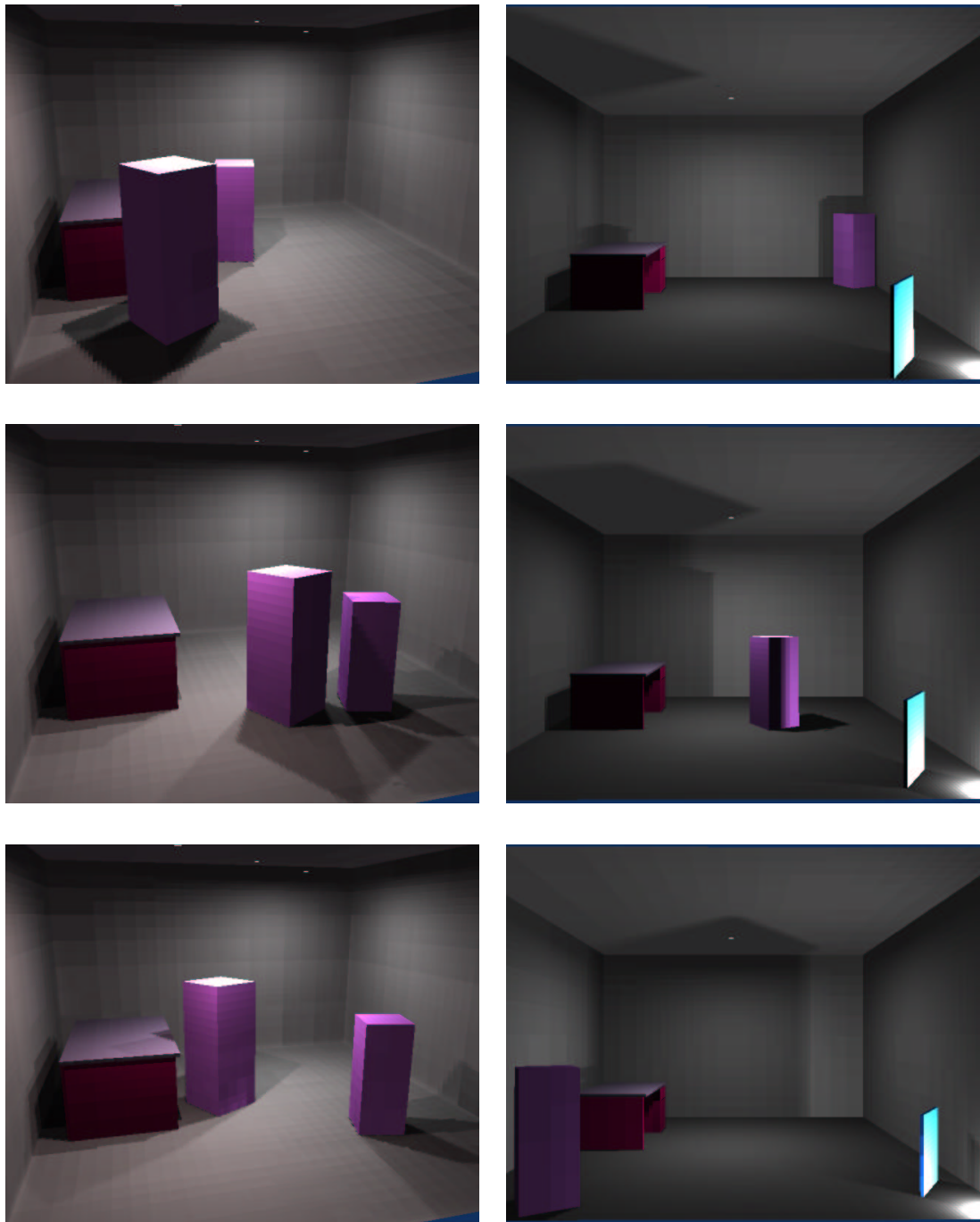
La construction d'une hiérarchie représentant une animation complète nécessite une importante quantité de mémoire. Le coût en est plus grand que pour le cas statique d'un facteur 10 (ACTEURS) et 17 (ÉCRAN). Il serait donc intéressant, comme nous le développerons au chapitre 8, de chercher à ordonner le raffinement de manière à pouvoir se décharger d'une partie de la hiérarchie sur disque.

Accélération des calculs

Notre algorithme accélère le calcul d'une animation, tout en produisant une illumination de même qualité. Les facteurs d'accélération sont de l'ordre de 2.5 (ACTEURS) et 2 (ÉCRAN). Ce résultat est quelque peu décevant si on le met en regard de l'augmentation de la consommation mémoire, et est clairement dû à notre utilisation intensive de l'échantillonnage temporel lors du raffinement.

Réduire le nombre d'échantillons entraîne une réduction conséquente du temps de calcul mais s'effectue aux dépens de la qualité de la solution : certains contours d'ombre peuvent alors être incorrectement détectés. Une technique de détermination de la visibilité plus évoluée, par exemple utilisant le volume englobant une trajectoire, devrait permettre d'obtenir une meilleure accélération.

Remarquons également que la scène ÉCRAN a été délibérément créée pour mettre notre technique en difficulté : l'illumination directe varie sur l'ensemble de la scène durant toute la durée de la séquence ; l'illumination indirecte en provenance du sol et des murs près de la source de lumière occultée est très importante et intervient dans des régions déjà finement subdivisées en raison du déplacement de l'ombre projetée par le panneau en mouvement. De très nombreux liens y sont donc créés, en raison du phénomène de « compétition » pour la subdivision entre les dimensions spatiales et temporelles que nous avons évoqué au paragraphe 4.3.



ACTEURS

ÉCRAN

FIG. 5.1: Quelques images représentatives des scènes sur lesquelles nous avons effectué nos tests. Nous les avons choisies pour illustrer des situations pouvant être rencontrées dans des applications pratiques. ÉCRAN présente des variations d'éclairage plus évidentes et à plus grande échelle (aussi bien en temps qu'en espace) dues au mouvement de l'obstacle devant la source de lumière située à droite au niveau du sol.

Hiérarchisation des échanges

Du fait de la nature hiérarchique de notre technique, le facteur d'accélération obtenu dépend de la profondeur de la hiérarchie, donc de la longueur de la séquence : dans le cas de la scène ACTEURS, doubler la durée de l'animation, tout en conservant un mouvement équivalent (ce qui équivaut à multiplier le nombre d'images par seconde par 2) ne provoque qu'une augmentation de 55% environ du temps de calcul et de la place mémoire nécessaire. Ce résultat prouve que, de même que le raffinement se concentre autour des bordures d'ombre dans le cas statique, notre algorithme ne produit de subdivisions supplémentaires que dans les zones où des détails temporels additionnels sont requis.

On notera que notre algorithme traite plus efficacement l'éclairage indirect : un facteur d'accélération de 4 (ACTEURS) et 3 (ÉCRAN) est observable si on ne prend en compte que le temps de calcul nécessaire à la simulation des échanges indirects. Ceci est dû au fait que les échanges indirects varient plus régulièrement dans le temps comme dans l'espace puisqu'ils ne génèrent pas de bordures d'ombres nettes.

5.2 Mise en évidence des points nécessitant amélioration

5.2.1 Calcul sur des scènes complexes

De même que l'algorithme de radiosit  hi rarchique statique, la complexit  de notre algorithme cro t de fa on quadratique avec le nombre de polygones composant originellement la sc ne (*cf.* paragraphe 1.2.4). L'oracle de raffinement doit en effet  tre appliqu    l'ensemble des paires de surfaces possibles pour initialiser l'algorithme.

Dans notre cas, cette  valuation du crit re de raffinement est particuli rement lente : notre oracle requiert d'autant plus d' chantillons temporels que l'intervalle de temps correspondant   l'interaction est long (*cf.* paragraphe 4.2.2). Puisque ces liens sont les premiers consid r s, ils ne sont pas encore divis s en temps. L' valuation du crit re de raffinement n cessitera donc le nombre maximum d' chantillons. Tel que nous l'avons pr sent  aux chapitres pr c dents, notre algorithme ne permet donc d'effectuer des calculs que sur une sc ne compos e d'au plus quelques centaines de polygones. Au del , le temps n cessaire au calcul d'une animation le rend en pratique inutilisable. De plus, la place m moire n cessaire au stockage des nombreux liens cr es devient rapidement d raisonnable.

5.2.2 Discontinuit s dans l' clairage indirect

Nous avons pr c demment insist  sur l'importance de la continuit  temporelle de l' clairage pour la qualit  de l'animation (*cf.* paragraphe 2.1.2). Il nous faut en effet prendre garde   ce que la s quence d'images produite soit exempte de variations brusques dans l' clairage.

Rappelons que des discontinuit s similaires peuvent  tre observ es dans le domaine spatial pour l'algorithme de radiosit  hi rarchique classique. Ces derni res ne sont pas uniquement dues   une pr cision insuffisante des calculs. Elles apparaissent notamment pour l' clairage indirect, lorsque l'oracle de raffinement employ  ne raisonne que sur l'erreur globalement commise dans l' valuation de la radiosit  sur l' l ment, sans prendre en compte sa r partition¹.

De plus, l'utilisation d'une base de fonctions uniformes ne permet pas une r partition homog ne de cette erreur sur l' l ment. Ceci entra ne, pour les  changes  nerg tiques de faible intensit , l'apparition de discontinuit s aux fronti res des  l ments. De plus, le m canisme de mise   jour hi rarchique fait que les discontinuit s dues aux  changes r partis entre les diff rents niveaux s'accumulent. Ainsi,

¹C'est, entre autres, le cas du crit re  nerg tique propos  par Hanrahan *et al.* [55] (crit re « BF »).

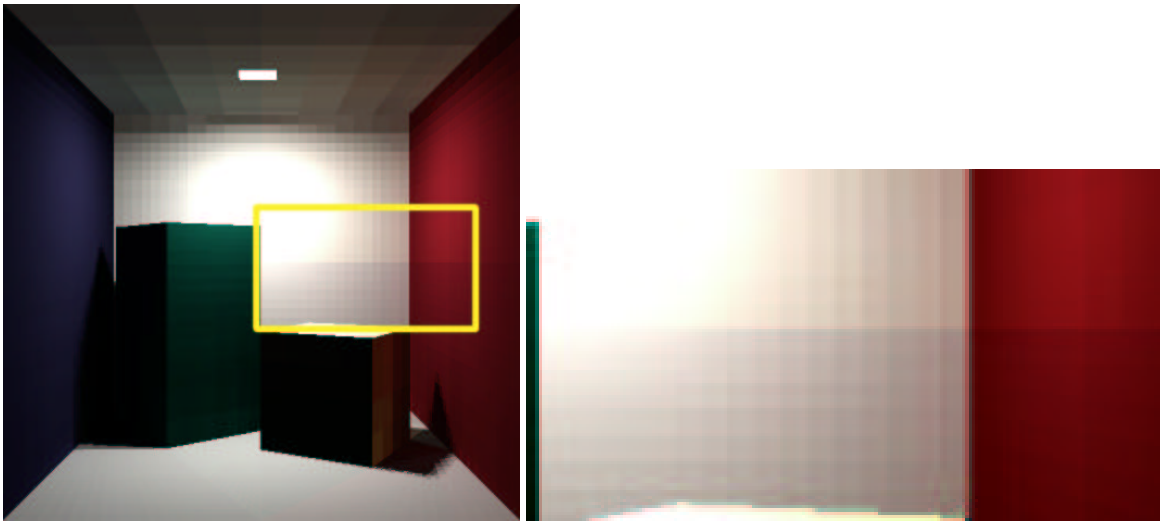


FIG. 5.2: Exemple de discontinuités spatiales.

on peut généralement observer d'importantes discontinuités aux frontières des éléments de grande taille situés aux premiers niveaux de la hiérarchie (cf. figure 5.2).

Dans certaines conditions, notre algorithme de radiosité hiérarchique spatiale et temporelle peut entraîner l'apparition de telles discontinuités dans la dimension temporelle. Tel que nous l'avons décrit, il utilise en effet une base de fonctions uniformes. De plus, l'oracle de raffinement que nous avons présenté ne se base que sur les variations du noyau géométrique de l'équation de radiosité. La répartition de l'erreur sur le récepteur due aux variations de la radiosité sur l'émetteur n'est donc pas prise en compte.

Si l'on construit une scène où l'éclairage est majoritairement d'origine indirecte et d'intensité variable (par exemple en déplaçant rapidement les sources de lumière), on voit donc apparaître d'importantes discontinuités temporelles aux dates correspondant aux principales subdivisions (c'est-à-dire au milieu de l'animation, ainsi qu'aux premier et second quarts). Ces discontinuités sont d'autant plus remarquées qu'elles affectent quasiment l'ensemble de la scène au même instant.

Afin de mettre en évidence l'existence de ces discontinuités, nous avons construit une scène pour laquelle notre algorithme se comporte de façon manifestement non satisfaisante. Elle est composée de quatre boîtes placées dans une pièce aux murs colorés (respectivement en rouge, vert, bleu et gris). Dans cette pièce se déplacent quatre projecteurs de lumière n'éclairant qu'une étendue limitée. Ainsi, la majorité de l'éclairage est indirect. De plus, alors que les projecteurs balayent les murs, la couleur et l'intensité de l'éclairage indirect changent radicalement. L'animation obtenue présente d'importantes discontinuités temporelles, qu'illustre la figure 5.3.

5.3 Conclusion

L'algorithme de radiosité hiérarchique spatiale et temporelle a pour objectif de calculer l'illumination globale pour des scènes animées, plus rapidement que si l'on procédait à une simulation image par image. Nous avons montré que, tel que nous l'avons présenté aux chapitres précédents, il permet d'accélérer de tels calculs sur des scènes géométriquement simple. Ceci démontre que le concept d'une hiérarchisation *temporelle* des échanges lumineux, afin d'en accélérer les calculs est conceptuellement valide.

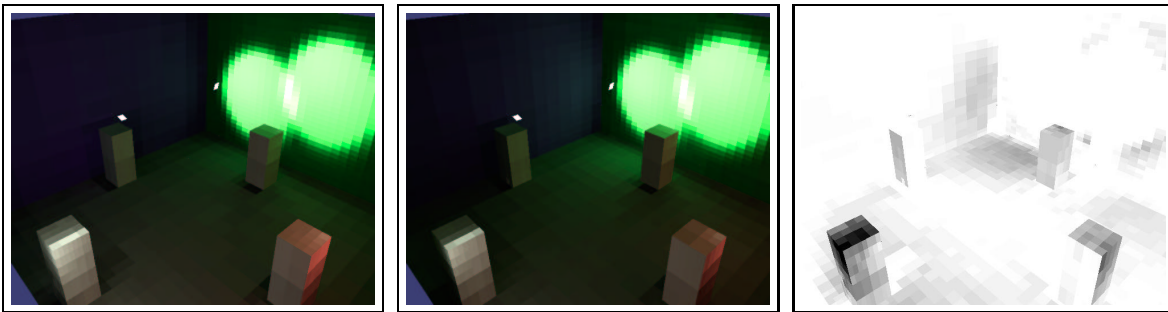


FIG. 5.3: Exemple de discontinuité temporelle. Nous montrons la scène aux dates $t = \frac{1}{2} - \epsilon$ et $t = \frac{1}{2} + \epsilon$, ainsi que leur différence (en fausses couleurs). On peut constater que l'illumination de toute la scène a été modifiée en l'espace d'une seule image.

Toutefois, nous avons montré que des obstacles subsistent, s'opposant à son utilisation sur des scènes plus complexes :

- De même que pour l'algorithme de radiosité hiérarchique statique, il est difficile de l'employer pour des scènes composées de plusieurs milliers de polygones. De tels calculs ne sont en pratique envisageables que si l'on étend la hiérarchie au-dessus du niveau des surfaces, de façon à réduire la complexité du problème posé. Or, la nature hiérarchique de cet algorithme le rend d'autant plus performant (par rapport à une approche non hiérarchique) que la profondeur de la hiérarchie est grande. Il devrait donc réaliser pleinement son potentiel sur des scènes complexes. Nous montrerons au chapitre 6 comment étendre à notre algorithme le concept de regroupement des surfaces (*cf.* paragraphe 1.2.4), afin de lever cet obstacle.
- Pour des scènes dont l'éclairage est majoritairement d'origine indirecte et d'intensité variable, des discontinuités apparaissent dans la représentation temporelle de l'éclairage. Or nos expériences montrent également que c'est précisément pour l'éclairage indirect que notre algorithme permet de factoriser le plus efficacement les calculs. Nous détaillerons, au chapitre 7, comment l'utilisation d'une base de fonctions d'ordre supérieur permet d'obtenir une représentation continue de la radiosité au cours du temps.
- Le coût mémoire de notre algorithme est important en regard de la complexité limitée des scènes sur laquelle nous avons effectué nos démonstrations. Nous montrerons, au chapitre 8, qu'il est possible de mettre à profit la « localité » temporelle des échanges lumineux (deux éléments dont les intervalles de temps sont disjoints n'échangent pas de lumière) afin de permettre de ne charger en mémoire qu'une partie de la hiérarchie en conservant l'essentiel des performances de notre algorithme.

Troisième partie

Mise en pratique de l'algorithme de radiosité hiérarchique spatiale et temporelle

Gestion de la complexité : regroupement des surfaces

UNE des principales faiblesses de l'algorithme de radiosité hiérarchique spatiale et temporelle que nous avons présenté dans la précédente partie est son incapacité à traiter en un temps raisonnable des scènes géométriquement complexes. De même que pour l'algorithme de radiosité hiérarchique statique, son coût croît asymptotiquement comme le carré du nombre de polygones formant originellement la scène. En effet, toutes les surfaces échangent potentiellement de la lumière. Il nous faut donc en examiner chaque paire pour déterminer si une interaction est possible. Cela est d'autant plus problématique dans notre cas que l'évaluation de l'oracle de raffinement pour ces nombreux liens initiaux est très longue, puisqu'ils ont pour intervalle de validité temporelle l'intégralité de l'animation.

Afin de rendre l'algorithme de radiosité hiérarchique utilisable en pratique, il est donc nécessaire de remédier à cette limitation. Nous avons vu au paragraphe 1.2.4 que l'extension de la hiérarchie au-dessus des surfaces sous forme de groupes [105, 99] permet d'éliminer la phase préliminaire de liaison des surfaces, en la remplaçant par l'établissement d'un lien unique de la racine de la hiérarchie vers elle-même. Si on fait l'hypothèse d'une répartition harmonieuse des liens [55, 95], le coût de l'algorithme de radiosité hiérarchique avec regroupement croît linéairement avec le nombre de surfaces composant initialement la scène.

Dans ce chapitre, nous montrons comment adapter la notion de groupe au cas de la radiosité hiérarchique spatiale et temporelle. Cette transposition n'est pas triviale. Généralement, la hiérarchie des groupes est construite dans une phase préliminaire. Dans notre cas, cela n'est pas acceptable. En effet, s'il existe une hiérarchie de groupes antérieure au processus de raffinement, la subdivision temporelle d'un lien ne sera possible qu'à partir du moment où celui-ci se situe au niveau des surfaces. Nous montrons, au paragraphe 6.1 comment il est possible d'obtenir la construction de la hiérarchie

de groupes par un effet de bord du processus de raffinement des liens¹. De plus, il nous faut déterminer comment gérer les interactions des groupes entre eux et avec les surfaces, en émission et en réception. Nous exposons au paragraphe 6.2 comment intégrer l'utilisation des groupes dans notre algorithme de radiosité spatiale et temporelle. Finalement, nous montrons les résultats que nous permet d'obtenir l'adjonction des groupes à notre algorithme.

6.1 Construction paresseuse de la hiérarchie de groupes

6.1.1 Principe de la construction

Nous souhaitons construire une hiérarchie de groupes prolongeant au-dessus du maillage des surfaces notre hiérarchie spatiale et temporelle. Ainsi, les relations hiérarchiques entre groupes parents doivent pouvoir être spatiales (un groupe physiquement inclus dans un autre) ou temporelles (un groupe ayant pour intervalle de validité une partie de celui du groupe père). Nous découplons, pour les groupes comme pour les surfaces, subdivisions spatiales et temporelles : un groupe est donc subdivisé soit spatialement, soit temporellement. En conséquence (cf. paragraphe 4.3), lorsqu'une subdivision *spatiale* a eu lieu, on ne peut lui superposer une subdivision temporelle. Or une hiérarchie de groupes, antérieure au processus de raffinement, formerait une hiérarchie purement spatiale au-dessus des surfaces. Le processus de raffinement ne pourrait donc procéder à une subdivision temporelle avant d'avoir atteint le niveau des surfaces.

Il paraît donc souhaitable de produire la hiérarchie des groupes non plus au cours d'une phase préliminaire mais, de même que le maillage hiérarchique, pendant la phase de raffinement des liens. Ainsi, on pourra décider, en fonction des variations locales de l'éclairage, si pour un groupe une subdivision spatiale est plus appropriée qu'une subdivision temporelle. Pour cela, il nous faut donc définir un mécanisme de subdivision hiérarchique similaire à celui employé pour les surfaces (cf. paragraphe 4.1.2). Nous procédons donc à un *regroupement paresseux* des surfaces, que nous ne n'effectuons qu'au moment du raffinement. En conséquence, la hiérarchie de groupes produite n'est complète que là où une grande précision dans les échanges est requise.

6.1.2 Critères pour le choix d'un algorithme de construction

Il nous faut, dans un premier temps, déterminer comment trier par proximité les surfaces dans un groupe, afin de produire un niveau supplémentaire dans la hiérarchie. De nombreux algorithmes ont été développés pour le regroupement hiérarchique des surfaces, que ce soit pour la production d'une structure d'accélération des requêtes de visibilité ou pour les besoins de la radiosité hiérarchique. Toutefois, tous ne peuvent pas être employés dans notre cas.

Certains de ces algorithmes produisent une hiérarchie de groupes forcement fixes dans l'espace, c'est-à-dire ne pouvant être attachés aux surfaces mobiles. C'est par exemple le cas de la hiérarchie de Kd-arbres employée par Sillion [99], et que nous avons modifiée dans [56]. Pour utiliser une telle structure, il nous faudrait permettre aux surfaces en mouvement de se déplacer dans un groupe (ce qui compliquerait inutilement le mécanisme de répartition de l'énergie incidente), et forcer une subdivision temporelle dès qu'une surface sort entre ou sort d'un groupe (cf. figure 6.1). Ces algorithmes ne nous paraissent donc pas appropriés dans notre cas. Nous préférons nous tourner vers les algorithmes produisant des groupes de boîtes englobantes. En effet, ces dernières peuvent « suivre » les objets en mouvement dans leur déplacement.

¹Rappelons que le maillage hiérarchique des surfaces est produit de cette façon.

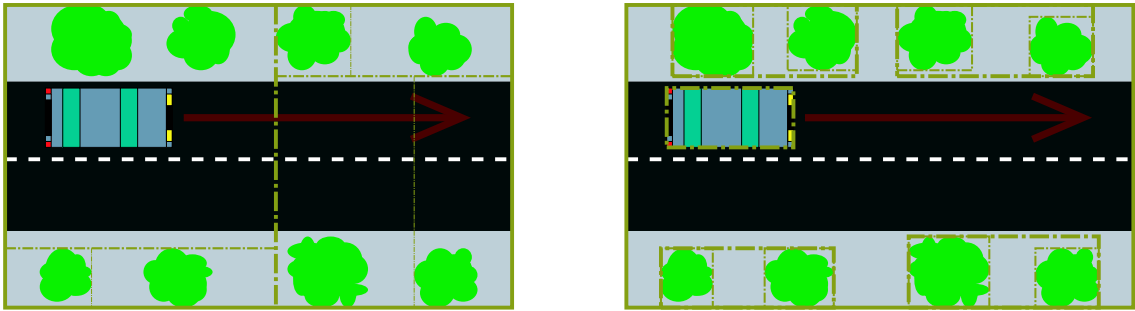


FIG. 6.1: Comparaison KD-arbre/boîtes englobantes. À gauche : Un objet en mouvement traversera tôt ou tard les frontières d'un KD-arbre. À droite : Une hiérarchie de boîtes englobantes peut épouser le mouvement d'un objet.

Comme nous souhaitons effectuer un regroupement paresseux, l'algorithme permettant le regroupement des surfaces doit pouvoir être appliqué de façon locale (c'est-à-dire au contenu d'un groupe donné et pas à l'ensemble de la scène). De plus, il doit pouvoir n'effectuer de regroupement que sur un seul niveau hiérarchique.

Certains algorithmes de regroupement évaluent l'utilité de l'ajout d'un objet à un groupe grâce à une fonction de coût (tel l'algorithme de Goldsmith *et al.* [42]). Nous ne pouvons les utiliser puisqu'ils procèdent de façon ascendante (approche *bottom-up*) : partant de l'ensemble des surfaces, ils construisent la hiérarchie de groupes en partant des feuilles, et en finissant par le sommet. Pour les mêmes raisons, l'algorithme présenté dans [56] et développé en annexe dans ce mémoire ne peut être utilisé dans notre cas puisqu'il procède à la construction des boîtes englobantes de façon ascendante.

Les algorithmes inspirés de celui de Goldsmith *et al.* sont généralement assez coûteux. Ils considèrent en effet la possibilité d'un regroupement pour un grand nombre de paires d'objets. Or, nous avons besoin d'effectuer le regroupement rapidement puisque cette fonction sera fréquemment appelée au cours du raffinement. Les méthodes procédant par insertion séquentielles des objets dans la hiérarchie nous paraissent plus prometteuses, puisque leur coût varie linéairement avec le nombre d'objets à regrouper.

À notre connaissance, seul l'algorithme proposé par Christensen *et al.* [21] permet la construction rapide d'une hiérarchie de boîtes englobantes, de façon descendante (approche *top-down*). Nous présentons au paragraphe suivant comment l'intégrer au mécanisme de subdivision spatiale d'un groupe.

Comme nous l'avons indiqué au paragraphe 4.2.3, nous utilisons également l'algorithme de Christensen *et al.* pour produire la structure d'accélération pour les requêtes de visibilité. Il n'est toutefois pas possible de faire l'économie de la construction de cette seconde structure et d'utiliser la hiérarchie de groupes pour cette tâche (comme le font de nombreuses implémentations de l'algorithme de radiosité hiérarchique). En effet la hiérarchie de groupes au début du processus de raffinement n'est qu'en partie construite. En pratique, cela cause un ralentissement inacceptable de l'évaluation des interactions au cours de la première itération. Il est donc préférable de construire, pour ces deux tâches, deux structures séparées. De plus, comme nous l'avons remarqué dans [56], les critères d'appréciation pour ces deux utilisations d'une hiérarchie de groupes ne sont pas nécessairement les mêmes. La construction de deux structures séparées permet des réglages différents (par exemple pour la tolérance au recouvrement, ou pour le facteur de branchement moyen désiré).

6.1.3 Subdivision spatiale d'un groupe

Pour effectuer la subdivision spatiale d'un groupe, nous procédons donc au regroupement des surfaces qu'il contient en fonction de l'octant dans lequel se situe leur centre de gravité (cf. figure 4.4). Il est possible également, si l'on souhaite réduire le facteur de branchement moyen de la hiérarchie, de regrouper les surfaces situées dans une même moitié (en subdivisant alternativement suivant chacun des axes principaux). De même que pour l'algorithme 4.3, nous ne permettons pas la création d'un groupe contenant seulement une surface. À la différence de l'algorithme 4.2, nous ne procédons pas à un appel récursif, afin de ne construire les groupes que pour un seul niveau hiérarchique. L'algorithme résultant est donné en pseudo-code dans l'encadré 6.1.

Algorithme 6.1 Mécanisme de subdivision spatiale d'un groupe (d'après [21])

Subdivision_spatiale(Données : groupe)

Variables :

 | *tableau_listes*[8]
 | *groupe_fils*

pour toute *surface* ∈ *Contenu*(groupe) **faire**

 | **si** (*Taille*(*surface*) ≥ *Taille*(groupe) × (1 + ε) / 2) **alors**

 | *Inclure*(*surface*, groupe)

 | **sinon**

 | *Ajouter*(*surface*, *tableau_listes* [*Octant*(*surface*)])

 | **fin si**

fin pour

pour tout *octant* ∈ {1, ..., 8} **faire**

 | **si** (*Longueur*(*tableau_listes*[*octant*] > 1) **alors**

 | *groupe_fils* ← *Boite_Englobante*(*tableau_listes*[*octant*])

 | *Inclure*(*groupe_fils*, groupe)

 | **sinon**

 | *Inclure*(*Unique_Surface*(*tableau_listes*[*octant*]), groupe)

 | **fin si**

fin pour

6.1.4 Subdivision temporelle d'un groupe

Afin de produire une hiérarchie mixte de groupes (spatiale *et* temporelle), il nous faut également permettre la subdivision temporelle des groupes. Pour cela, il suffit de dupliquer le groupe à subdiviser ainsi que son contenu, comme l'illustre la figure 6.2. Les deux groupes fils (ainsi que les surfaces qu'ils contiennent) auront pour intervalles de validité les deux intervalles obtenus par subdivision de celui du groupe père. Afin de réduire la place mémoire nécessaire au fonctionnement de l'algorithme, comme on l'a noté au paragraphe 4.1.2, il est possible de mettre en commun une grande partie des données entre le groupe père et ses fils : ne doivent effectivement être créés que leurs descriptions temporelles, et leurs radiosités. Toutes les autres données (en particulier leur description géométrique, ainsi que celle de leur trajectoire) peuvent être partagées.

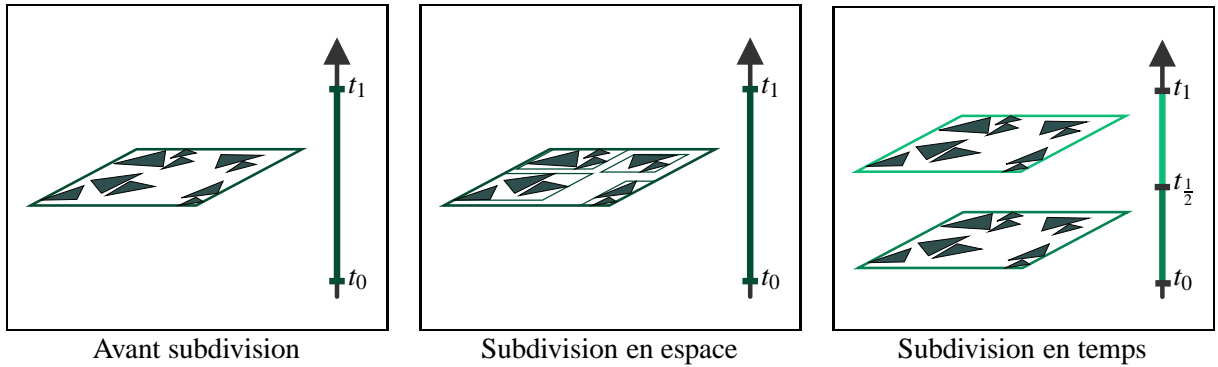


FIG. 6.2: Le groupe représenté à gauche (en 2D) peut être subdivisé spatialement par regroupement des surfaces qu'il contient (au centre), ou bien temporellement (à droite) par duplication et subdivision de son intervalle de validité.

Algorithme 6.2 Mécanisme de subdivision temporelle d'un groupe

Subdivision_temporelle(Données : groupe)

Variables :

 groupe_fils_1, groupe_fils_2
 surface_soeur

pour toute surface \in Contenu(groupe) **faire**

 Partager_geom(groupe, groupe_fils_1)
 Partager_geom(groupe, groupe_fils_2)
 Partager_trajectoire(groupe, groupe_fils_1)
 Partager_trajectoire(groupe, groupe_fils_2)
 Partager_geom(surface, surface_soeur)
 Partager_trajectoire(surface, surface_soeur)
 groupe_fils_1.intervalle \leftarrow **Premiere_Moitie**(groupe.intervalle)
 groupe_fils_2.intervalle \leftarrow **Seconde_Moitie**(groupe.intervalle)
 Partager_intervalle(groupe_fils_1, surface)
 Partager_intervalle(groupe_fils_2, surface_soeur)
 Retirer(surface, groupe)
 Inclure(surface, groupe_fils_1)
 Inclure(surface_soeur, groupe_fils_2)
 Inclure(groupe_fils_1, groupe)
 Inclure(groupe_fils_2, groupe)

fin pour

6.2 Utilisation des groupes

Après avoir détaillé comment obtenir une hiérarchie spatiale et temporelle de groupes, il nous reste à préciser comment l'employer. En effet, les mécanismes de gestion des échanges lumineux en provenance ou à destination d'un groupe (cf. paragraphe 1.2.4) sont plus complexes que ceux employés pour les surfaces. Notre approche est basée sur celle présentée par Sillion dans le cas statique [100]. Il s'agit, pour simplifier, de prendre en compte le caractère anisotrope d'un groupe en émission, en évaluant les termes directionnels directement au niveau des surfaces (cf. paragraphe 6.2.1).

De même, l'irradiance reçue par un groupe à travers un lien sera distribuée immédiatement à toutes les surfaces qu'il contient conformément à leur orientation (cf. paragraphe 6.2.2). Nous discuterons également au paragraphe 6.2.3 différentes possibilités pour le choix de la méthode d'échantillonnage spatial des groupes.

6.2.1 Émission depuis un groupe

Il apparaît très difficile de définir une quantité géométrique équivalente à la notion de facteur de forme dans le cas des groupes. En effet, comme nous l'avons fait remarquer au paragraphe 1.2.4, on ne peut pas considérer un groupe comme étant un émetteur isotrope. Il nous faut donc calculer directement la puissance lumineuse émise en direction du récepteur. Malheureusement, si nous calculions exactement la contribution de chaque surface qu'un groupe contient, pour en évaluer l'émission dans une direction, nous violerions le principe de hiérarchisation des calculs. Les performances de l'algorithme s'en ressentirait de façon inacceptable.

Or, la résolution des requêtes de visibilité représente la majeure partie du temps de calcul de notre algorithme de radiosité hiérarchique. En pratique, le coût de l'évaluation des termes angulaires et des distances peut être considéré comme négligeable en regard de celui de l'échantillonnage de la fonction de visibilité v . Nous pouvons donc nous permettre d'effectuer directement au niveau des surfaces le calcul des différents termes angulaires afin de prendre en compte les différentes orientations des surfaces incluses. En revanche, l'évaluation de la visibilité doit être « factorisée », c'est à dire effectuée au niveau du groupe conteneur.

L'irradiance reçue par un récepteur P en provenance d'un émetteur Q est, dans le cas d'une base de fonctions uniformes par morceaux, la moyenne sur ce récepteur de l'irradiance reçue par chacun des points p de P pendant l'intervalle de temps où il est défini :

$$I_{recue} = \iint_P I_{recue}(p, t) dp dt$$

En reprenant les notations employées au chapitre 3, exprimons l'irradiance totale reçue en un point p en provenance du groupe Q comme la somme des irradiances émises par les surfaces Q_i qu'il contient :

$$I_{recue}(p, t) = \sum_i \int_{Q_i} B_{Q_i}(p, t) k(p, q, t) v(p, q, t) dq$$

Si l'on factorise la visibilité au niveau du groupe, en la remplaçant par un terme d'atténuation approximatif $\tilde{V}(t)$ (fonction du temps seulement), on peut approcher l'irradiance reçue par :

$$\tilde{I}_{recue}(p, t) = \tilde{V}(t) \sum_i \int_{Q_i} B_{Q_i}(p, t) k(p, q, t) dq dt \quad (6.1)$$

avec

$$\tilde{V}(t) = \frac{1}{A_P A_Q} \int_P \int_Q v(p, q, t) dp dq \quad (6.2)$$

Dans le cas où le récepteur P est une surface, on sait que la fonction k prend la forme suivante :

$$k(p, q, t) = \frac{\cos(\theta(p, q, t)) \cos(\theta(q, p, t))}{\pi r(p, q, t)}$$

Dans le cas où il s'agit d'un groupe, le terme directionnel du récepteur $\cos(\theta(p, q, t))$ sera pris en compte plus tard, lors de la phase de mise à jour de la hiérarchie que nous décrirons au paragraphe suivant. Ainsi, la forme générale du noyau géométrique k devient (en utilisant la notation de [100]) :

$$k(p, q, t) = \frac{\mathcal{R}(p, q, t) \cos(\theta(q, p, t))}{\pi r(p, q, t)^2}$$

\mathcal{R} étant le facteur de réception défini par :

$$\begin{aligned} \mathcal{R}(p, q, t) &= \cos \theta(p, q, t) && \text{si le récepteur est une surface} \\ &= 1 && \text{s'il s'agit d'un groupe} \end{aligned}$$

Il nous reste à définir comment intégrer l'équation 6.1 sur l'élément récepteur pour calculer la projection de l'irradiance reçue sur la base de fonctions utilisée. Si l'on utilise une base de fonctions uniformes par morceaux, cela revient simplement à échantillonner la fonction \tilde{I}_{recue} pour différents points et à différentes dates et à en calculer la moyenne. Nous présenterons et discuterons au paragraphe 6.2.3 les politiques d'échantillonnage spatial et temporel que nous avons employées.

6.2.2 Réception de l'énergie dans un groupe

Lorsque l'émetteur est une surface, l'irradiance reçue par un groupe est calculée par le biais de l'évaluation d'un « facteur de forme » d'une surface vers un groupe défini de la façon suivante :

$$\mathcal{F}_{P,Q} = \frac{1}{4A_P(t_P^{final} - t_P^{initial})} \iiint_{P,Q} \frac{\mathcal{R}(p, q, t) \cos(\theta(q, p, t))}{\pi r(p, q, t)^2} v(p, q, t) dq dp dt \quad (6.3)$$

où A_P est l'aire équivalente du groupe P , égale à la somme des aires des surfaces incluses dans le groupe P ou ses fils.

On remarquera que, de même qu'au paragraphe 6.2.1, le terme angulaire $\cos(\theta(p, q, t))$, fonction de l'orientation du récepteur, a été remplacé par le facteur de réception $\mathcal{R}(p, q, t)$ (qui vaut ici 1), et sera pris en compte suivant un mécanisme que nous allons à présent exposer.

Nous avons décrit comment calculer l'irradiance que recevrait un groupe si l'on faisait abstraction des orientations des surfaces qu'il contient. Il nous faut distribuer cette irradiance à toutes les surfaces contenues dans le groupe au moment de la phase de collecte à travers les liens (cf. paragraphe 1.2.3), sans attendre la phase de mise à jour hiérarchique, en la modulant par un terme permettant de prendre en compte l'orientation relative à la direction d'incidence de la lumière. Comme cette orientation dépend du temps, il nous faut procéder, une fois encore, à une projection sur notre base de fonctions constantes. Chaque surface P_i contenue dans le groupe récepteur P doit donc se voir distribuer l'irradiance $I_i = \beta_i I_{recue}$, avec :

$$\beta_i = \int_{t_i^{initial}}^{t_i^{final}} \frac{\langle \vec{N}(p_i, t), (M(q, t) - M(p_i, t)) \rangle}{\|M(q, t) - M(p_i, t)\|} dt \quad (6.4)$$

où p_i et q sont les centres de gravité respectifs de la surface P_i et de l'émetteur Q , et les fonctions \vec{N} et M indiquent les normales et trajectoires des points de la scène (cf. paragraphe 3.1 et la figure 3.1). Nous effectuons le calcul de cette intégrale en utilisant une simple règle de quadrature gaussienne (cf. [88] pages 147 à 157).

6.2.3 Échantillonnage des groupes

Nous avons vu dans les paragraphes précédents que, comme pour les surfaces, l'on fait un usage abondant de l'échantillonnage spatial pour évaluer les échanges lumineux impliquant un ou plusieurs groupes. On peut envisager deux façons de procéder à cet échantillonnage : soit on fait abstraction du contenu du groupe (les surfaces) et on le considère comme un volume homogène, en plaçant des échantillons suivant une grille régulière, soit on cherche à générer des échantillons appartenant nécessairement aux surfaces formant le groupe (cf. figure 6.3).

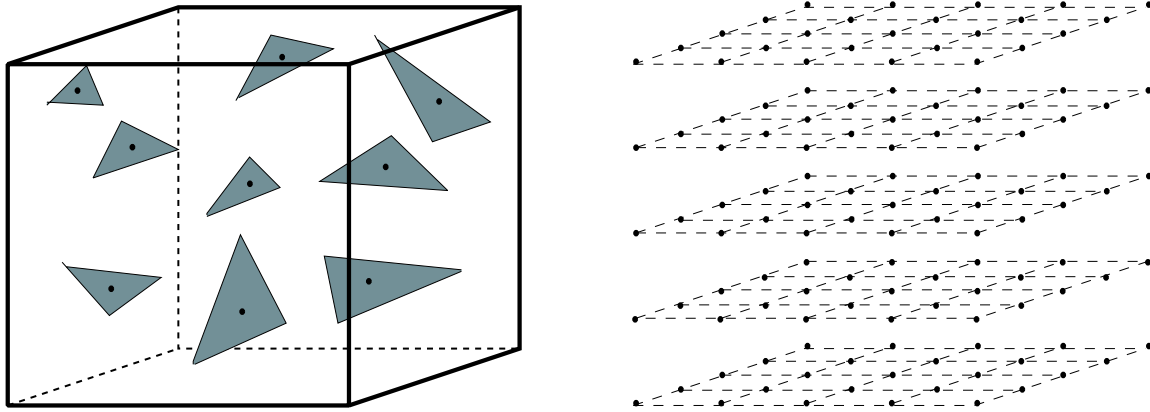


FIG. 6.3: Échantillonnage spatial d'un groupe : **à gauche**, sur les surfaces, **à droite**, suivant une grille régulière.

La première de ces deux approches est incontestablement plus rapide : il suffit en effet de pré-calculer un ou plusieurs ensembles de positions d'échantillons dans le cube unité que l'on « ajustera » à la taille du groupe considéré. On peut choisir de répartir ces échantillons à l'intérieur du volume ou à la surface de la boîte englobante. Si l'on dispose ces points suivant une grille régulière, leur nombre sera nécessairement une puissance de 3. En revanche, si l'on utilise un schéma de répartition pseudo-aléatoire (telle une séquence de Halton [54]), le nombre d'échantillons peut être quelconque.

La seconde approche est généralement plus lente : il faut parcourir récursivement tous les sous-groupes afin de recenser les surfaces qu'il contient, et ensuite parcourir la liste de surfaces ainsi formée pour placer les échantillons. Le coût de l'échantillonnage est alors de l'ordre du nombre de surfaces incluses dans le groupe. Toutefois, cette approche permet généralement une plus grande précision, et est parfois la seule envisageable (par exemple si l'on souhaite échantillonner la fonction \vec{N} qui n'est définie que sur les surfaces).

Nous avons choisi, dans le cas de l'intégrale de la visibilité 6.2, de sélectionner les échantillons sur une grille régulière car, dans ce cas, le nombre d'échantillons pris doit être indépendant du nombre de surfaces contenues dans le groupe afin de ne pas enfreindre le principe de hiérarchisation (cf. paragraphe 6.2.1). En revanche, dans le cas du calcul de l'irradiance en provenance d'un groupe (équation 6.1), nous sommes contraints de prendre les échantillons sur chaque surface, puisque nous devons avoir accès à la normale au point échantillon ainsi qu'à la radiosité de l'élément contenant ce point.

6.3 Résultats et discussion

Nous avons testé notre algorithme sur un ensemble de scènes de test (*cf.* figure 6.4) composées de plusieurs milliers de polygones, pour lesquelles le calcul d'une solution par notre méthode sans l'utilisation de groupes aurait été impossible.

La première scène se déroule dans une petite pièce meublée par quelques fournitures (bureaux, chaises, *etc.*). Une étagère, posée contre le mur, se renverse sur le sol. Cette animation est composée de 100 images, et dure 4 secondes. La scène est composée de 7200 polygones.

La seconde scène se déroule dans un grand hall meublé comprenant plusieurs bureaux séparés par des rayonnages de bibliothèques, et éclairé par de nombreuses lampes situées au plafond. Dans ce hall se déplace un personnage. Cette animation est composée de 500 images et dure 20 secondes. La scène est composée de 35000 polygones environ.

Nous comparons dans le tableau 6.1 les ressources nécessaires pour le calcul de ces solutions avec celles requises pour le calcul d'une solution image par image (par un algorithme de radiosité hiérarchique avec regroupement utilisant un critère de raffinement spatial équivalent). Ces temps de calculs ont été obtenus sur un processeur MIPS R12000 cadencé à 300 Mhz.

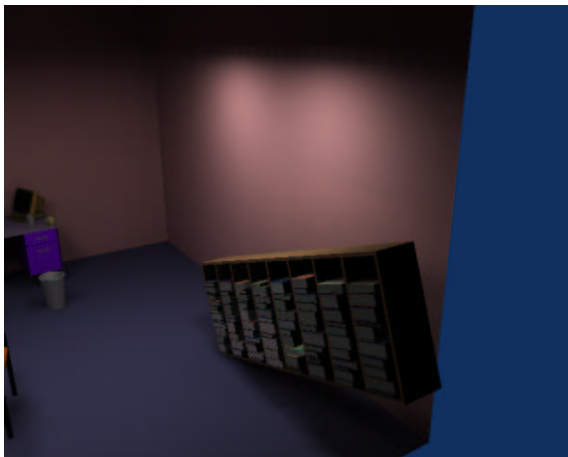
	Temps de calcul		Mémoire utilisée	
	Notre algorithme	Radiosité «statique»	Notre algorithme	Radiosité «statique»
ÉTAGERE	3.335 s	$184 \text{ s} \times 100 = 18.400 \text{ s}$	100 Mo	16 Mo
BUREAUX	33.333 s	$1.185 \text{ s} \times 500 = 592.500 \text{ s}$	842 Mo	120 Mo

TAB. 6.1: Résultats comparatifs pour l'utilisation du regroupement : sont indiqués le temps de calcul et la place mémoire requise par notre algorithme et par l'algorithme de radiosité hiérarchique avec regroupement statique employé pour une seule image.

On notera qu'effectuer des calculs sur des scènes comprenant initialement de nombreux polygones ne consomme pas nécessairement plus de mémoire que d'effectuer des calculs sur des scènes géométriquement plus simples. En effet, plus le nombre de polygones initiaux est grand, plus il est probable que ces derniers aient une aire faible. Par la suite, ils sont donc moins subdivisés spatialement. La taille de la hiérarchie produite par notre algorithme dépend essentiellement du nombre et de la puissance des sources de lumières principales de la scènes, ainsi que de l'échelle des variations d'éclairage au cours de l'animation. Le nombre de polygones initial n'influence pratiquement que le temps de calcul de la solution (puisque les requêtes de visibilité y sont sensibles).

On constate que nous obtenons pour l'ensemble de ces scènes des facteurs d'accélération beaucoup plus importants que ceux exhibés au chapitre précédent. Ceci peut s'expliquer par le fait que plus le nombre d'éléments dans la scène est grand, plus l'utilisation de la radiosité hiérarchique est avantageuse. L'utilisation des groupes, en permettant d'effectuer des calculs sur des scènes plus complexes permet de pleinement exploiter le potentiel de la formulation hiérarchique. De plus dans notre seconde scène, pour laquelle le facteur d'accélération est d'environ 18, les changements dans l'éclairage dus au mouvement du personnage sont assez localisés dans l'espace. Cela permet d'expliquer en partie cette bonne performance.

Nous obtenons des images de qualité similaire à celle obtenue pour les calculs image par image. La figure 6.4 montre quelques images des résultats obtenus pour chacune de nos scènes de test.



ÉTAGÈRE

BUREAUX

FIG. 6.4: Scènes de tests pour le regroupement

Traitement des discontinuités temporelles : utilisation de multi-ondelettes

Nous avons précédemment insisté sur l'importance de la continuité temporelle des animations produites par synthèse d'images (*cf.* paragraphe 2.1.2). Les ruptures dans la continuité de l'animation, tels que des sauts dans l'éclairage, de brusques changements dans les ombres, *etc.*, sont facilement repérée. Or, nous avons constaté que, dans le cas de scènes majoritairement éclairées de façon indirecte, et où les conditions d'éclairage varient grandement au cours du temps, des discontinuités temporelles pouvaient résulter de l'utilisation de notre algorithme de radiosité hiérarchique spatiale et temporelle (*cf.* paragraphe 5.2.2). Ces discontinuités *temporelles* sont d'une nature similaire aux discontinuités *spatiales* que l'on peut observer dans des conditions similaires sur des solutions calculées par l'algorithme de radiosité hiérarchique classique. Toutes sont causées par l'utilisation d'une base de fonctions uniformes pour la discrétisation de l'équation de la radiosité.

Afin de remédier à ce problème, nous transposerons dans le contexte des animations les solutions proposées dans la littérature pour le traitement des discontinuités dans le cas spatial. En particulier, l'utilisation des ondelettes nous est apparu comme prometteuse pour la représentation des variations temporelles de la radiosité, puisqu'elles s'intègrent naturellement dans le cadre de l'algorithme de radiosité hiérarchique. De plus, comme nous le verrons par la suite, les obstacles à leur utilisation rencontrés dans le cas spatial, que nous avons détaillés au paragraphe 1.2.5 (consommation de mémoire excessive, problème de définition sur un support générique, difficulté de concilier leur utilisation avec celle d'une hiérarchie de groupes, *etc.*) disparaissent dans le cas temporel.

Dans ce chapitre, nous montrerons comment améliorer la continuité temporelle des solutions produites par l'algorithme de radiosité hiérarchique spatiale et temporelle. Pour ce faire, nous utiliserons

une base de multi-ondelettes pour la représentation des variations temporelles de la radiosité sur les éléments du maillage. Dans un premier temps nous rappellerons le principe de construction de la base des multi-ondelettes de Alpert [3, 4] et leurs propriétés. Puis, nous détaillerons au paragraphe 7.2 comment nous avons employé cette base pour notre algorithme. Nous nous intéresserons à la description des changements à apporter à l'algorithme originel concernant le calcul des échanges lumineux entre objets. Nous y présenterons également un algorithme de raffinement adapté à notre nouvelle base. Cette dernière étant plus précise, son utilisation permet en effet de produire un maillage moins dense. Nous expliciterons ensuite les opérations à effectuer pour maintenir la cohérence entre les différents niveaux de détail. Finalement, au dernier paragraphe, nous présenterons et commenterons les résultats obtenus.

7.1 Ondelettes et multi-ondelettes

7.1.1 Analyse multirésolution

La théorie des ondelettes est un outil mathématique permettant la décomposition hiérarchique de fonctions. Originellement appliquée au traitement du signal, elle permet de combler certaines insuffisances de l'analyse de Fourier. En effet les décompositions en ondelettes permettent d'effectuer une étude des signaux à la fois en fréquence et en temps. Les ondelettes permettent ainsi d'effectuer une décomposition d'un signal avec une plus grande précision locale aux endroits souhaités, tout en permettant une approximation plus grossière pour les parties du signal jugées peu intéressantes.

Cette théorie offre ainsi un cadre mathématique naturel à l'étude et au développement d'algorithmes hiérarchiques tels que celui que nous avons présenté dans les chapitres précédents. De fait, Gortler *et al.* [46] ont montré que l'algorithme de radiosité hiérarchique de Hanrahan *et al.* [55] peut être décrit formellement comme une méthode basée sur une décomposition en ondelettes de l'équation de la radiosité. Nous allons rappeler brièvement ici les concepts de base de l'analyse multirésolution dans $\mathcal{L}_2([0, 1])$, (l'espace de Hilbert des fonctions dont le carré est intégrable) en nous appuyant sur l'exemple de la construction de la base de Haar, la plus simple qui soit.

Exemple des ondelettes de Haar

Si l'on subdivise régulièrement $[0, 1]$ en 2^j intervalles, on peut définir le sous-espace V^j des fonctions de $\mathcal{L}_2([0, 1])$ uniformes par morceaux sur chacun des intervalles obtenus. Les sous-espaces V^j (de dimension 2^j) sont emboîtés (*i.e.* $\forall j, V^j \subset V^{j+1}$) et l'union des V_j est dense dans $\mathcal{L}_2([0, 1])$ (c'est-à-dire que toute suite de fonctions dans l'union des V_j tend vers $\mathcal{L}_2([0, 1])$). Une base de V^j est formée par translation et homothétie de la fonction ϕ indicatrice de $[0, 1]$:

$$\phi_i^j(t) = \phi(2^j t - i)$$

On appellera les ϕ_i *fonctions d'échelle*. La figure 7.1 montre les fonctions d'échelle pour la résolution $j = 2$.

On appelle *espace de détail* W^j le complément orthogonal de V^j dans V^{j+1} . On appelle *ondelettes* les fonctions d'une famille $(\psi_i^j)_{i \in [0, 2^j - 1]}$ formant une base de W^j . Par construction, toutes les fonctions de V^{j+1} peuvent donc être décomposées en une combinaison linéaire des fonctions $\phi^0 = \phi$ et ψ_i^k avec $k \leq j$. La figure 7.2 montre les ondelettes de Haar pour la résolution $j = 1$.

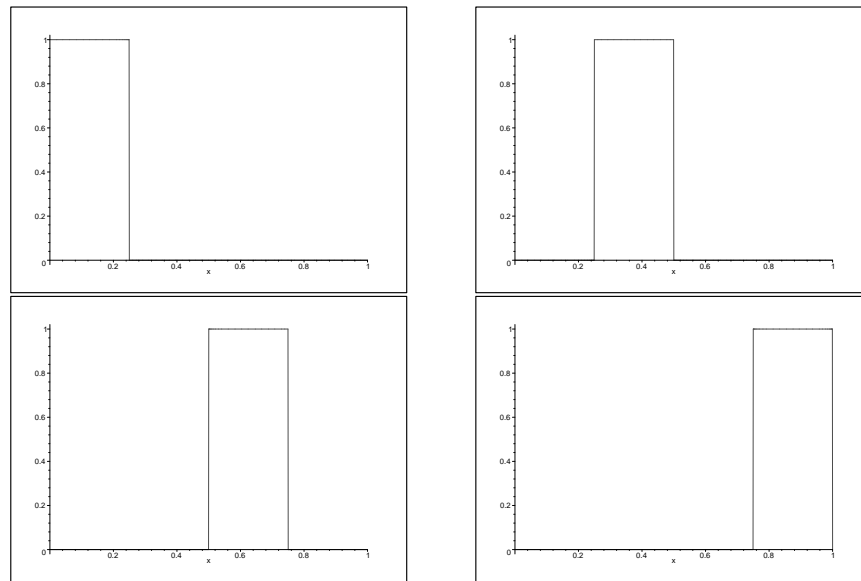


FIG. 7.1: Fonctions d'échelle pour la base de Haar de résolution 2

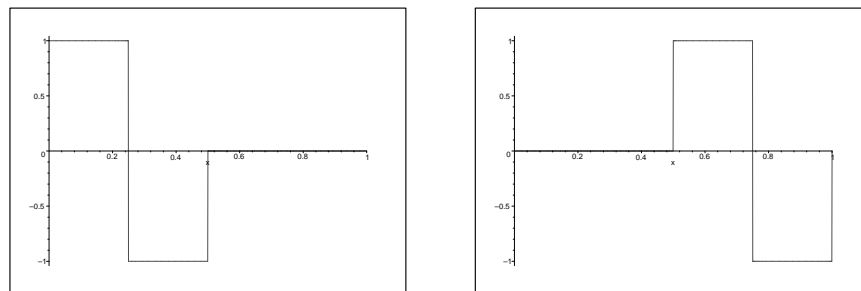


FIG. 7.2: Ondelettes de Haar de résolution 1

Propriétés des ondelettes

Toutes les bases d'ondelettes sont construites suivant un principe similaire à celui que nous venons d'exposer (choix d'une fonction d'échelle ϕ permettant de construire toutes les ϕ_i^j , puis construction des ondelettes par différence entre les ϕ_i^{j+1} et les ϕ_i^j). Parmi les bases d'ondelettes que l'on peut construire, il nous faut choisir en fonction de certaines propriétés que l'on peut juger désirables.

- **Moments nuls** : une ondelette ψ possède M moments nuls si et seulement si

$$\forall m \in [0, M - 1], \int \psi(t) t^m dt = 0$$

La base de Haar précédemment décrite, ne possède qu'un moment nul. Plus le nombre de moments nuls d'une famille d'ondelettes est grand, et plus les coefficients de la décomposition d'une fonction sur cette base vont décroître rapidement, donc meilleure sera une approximation composée de peu de niveaux de détail. Dans le cas particulier de la méthode de radiativité, un plus grand nombre de moments nuls nous garantit que la matrice du système linéaire du système à résoudre comportera plus de coefficients d'amplitude très faible, donc permet de réduire le nombre d'interactions à considérer [95].

- **Orthogonalité** : l'orthogonalité des ψ_i^j à l'intérieur d'un même niveau de détail j ne fait pas partie de la définition d'une base d'ondelettes. Pourtant, cette propriété est souhaitable car elle traduit une bonne séparation des détails apportés par chaque fonction.

- **Localité** : une fonction d'échelle est dite à *support compact* si l'ensemble des points pour lesquels elle prend une valeur non nulle est compact. De plus, une famille de fonctions d'échelle est dite à *support disjoint* si les supports de ces fonctions ont une intersection de mesure nulle. (Cette propriété entraîne automatiquement l'orthogonalité de la base d'ondelettes). Dans le cas de la méthode de radiosité, cette condition est quasiment obligatoire. En effet, il est extrêmement difficile de donner un sens aux « débordements » d'énergie d'un élément sur son voisin qui se produiront à la frontière entre deux éléments de maillage si elle n'était pas vérifiée.
- **Continuité et différentiabilité** : il est clair que plus les ondelettes seront « lisses », et plus elles permettront d'approximer précisément une fonction lisse. Dans le cadre de la représentation des variations temporelles de la radiosité, il est aisé de repérer les discontinuités d'ordre 0 (dont le traitement est notre objectif premier dans ce chapitre). En revanche, nous ne savons pas dans quelle mesure il est possible de percevoir les discontinuités d'ordre supérieur.

Malheureusement, ces critères d'appréciation sont contradictoires. Plus spécifiquement, la recherche d'un grand nombre de moments nuls et d'une plus grande continuité se font au détriment de la localité et de l'orthogonalité de la base d'ondelettes produite. En pratique, il est difficile de construire une base d'ondelettes orthogonales possédant plus d'un moment nul.

La première base d'ondelettes orthogonales à support compact a été construite par Daubechies [28]. Bien que continues, elles ne sont pas d'aspect lisse, comme l'illustre la figure 7.3. Il sera en pratique difficile de les employer pour la représentation temporelle de la radiosité. En effet, pour obtenir une fonction d'apparence constante sur un certain intervalle de temps (situation fréquente, *cf.* paragraphes 2.1.1 et 3.4), un raffinement extrêmement poussé sera nécessaire.

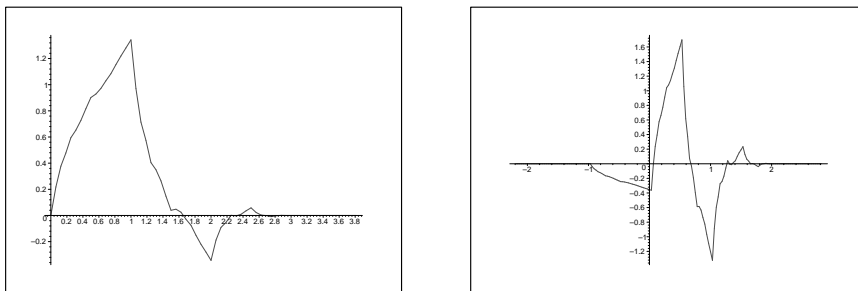


FIG. 7.3: Base de Daubechies D_4 : à gauche fonction d'échelle, à droite : ondelette

Nous verrons au paragraphe suivant qu'en autorisant l'emploi de plusieurs « formes » de fonctions pour la construction des fonctions d'échelle, il est possible d'obtenir de manière simple des bases de fonctions aux propriétés similaires à celles d'une base d'ondelettes, orthogonales, à supports disjoints, et possédant plusieurs moments nuls.

7.1.2 Les multi-ondelettes : une base orthogonale de $\mathcal{L}_2([0, 1])$

Nous allons dans ce paragraphe présenter rapidement les étapes amenant à la conception des multi-ondelettes de Alpert [4]. La différence majeure entre ces bases et les bases d'ondelettes classiques est qu'ici les fonctions d'échelle seront construites à partir de M fonctions génératrices $\phi^{m,0}$ différentes au lieu d'une seule fonction ϕ_0 .

Comme nous l'avons exposé au paragraphe précédent, nous souhaitons construire une base orthogonale. Afin que les différentes fonctions d'échelle soient orthogonales entre elles, il nous suffit de construire les $\phi^{m,0}$ orthogonales. De plus, nous souhaitons construire notre base de telle sorte qu'elle possède M moments nuls (*i.e.* que les fonctions de bases soient orthogonales à tous les polynômes de degré $M - 1$). Cette condition est aisément vérifiée si les $\phi^{m,0}$ sont des polynômes de degré m orthogonaux deux à deux, avec $m \leq M - 1$.

Par exemple, pour construire la base \mathcal{M}_2 (multi-ondelettes avec deux moments nuls, donc linéaires par morceaux), nous devons construire deux polynômes $\phi^{0,0}$ et $\phi^{1,0}$ sur $[0, 1]$, respectivement constant et de degré 1, et orthonormés. Puisque $\|\phi^{0,0}\| = 1$, il est clair que $\phi^{0,0} = 1$ sur $[0, 1]$. On a donc :

$$\begin{cases} \phi^{1,0}(t) & = \alpha t + \beta \\ \int_0^1 \phi^{1,0}(t) dt & = 0 \\ \int_0^1 \phi^{1,0}(t)^2 dt & = 1 \end{cases}$$

d'où l'on déduit rapidement que $\phi^{1,0}(t) = \pm\sqrt{3}(2t - 1)$. Puis de la même façon que nous avons procédé pour la base de Haar, nous définissons les fonctions d'échelle pour les différents niveaux de détail, par translation et homothétie des fonctions génératrices $\phi^{0,0}$ et $\phi^{1,0}$:

$$\phi_i^{0,j}(t) = \phi^{0,0}(2^j t - i) \text{ et } \phi_i^{1,j}(x) = \phi^{1,0}(2^j t - i)$$

De même, les fonctions de détail sont obtenues par différence entre les niveaux hiérarchiques. La figure 7.4 montre la base orthonormale correspondant au niveau de détail 0 (les deux fonctions d'échelle et les deux fonctions de détail associées).

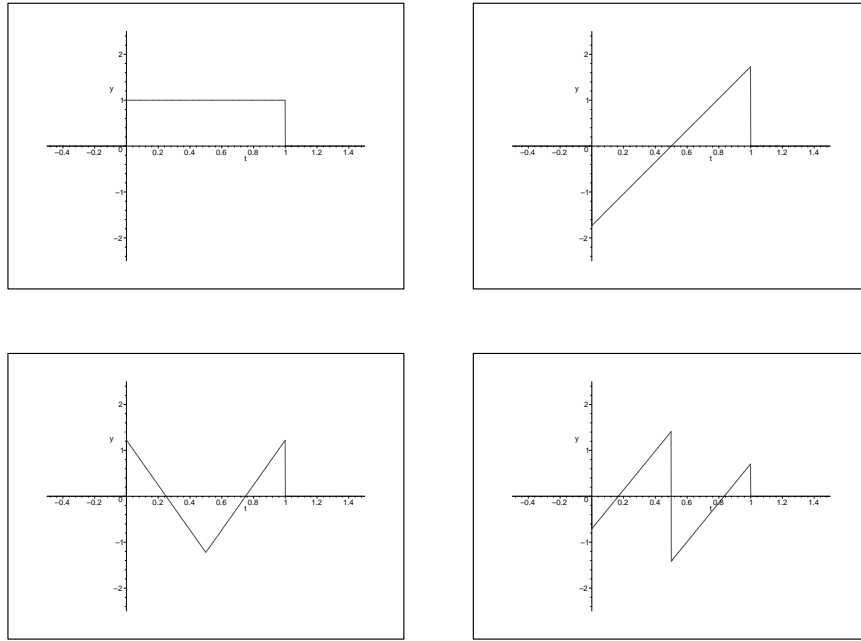


FIG. 7.4: Multi-ondelettes \mathcal{M}_2 : **En haut** fonctions d'échelle, **En bas** : ondelettes

Plus généralement, le processus de construction des fonctions $\phi^{m,0}$ que nous avons utilisé est lié à la définition de la base orthogonale (sur $[-1, 1]$) des polynômes de Legendre P_m , définis par la récurrence suivante :

$$\begin{cases} P_0(t) & = 1 \\ P_1(t) & = x \\ P_{m+1}(t) & = \frac{2m+1}{m+1}xP_m(x) - \frac{m}{m+1}P_{m-1}(x) \end{cases}$$

On a, après relocalisation sur l'intervalle $[0, 1]$ et normalisation :

$$\forall m, \forall t \in [0, 1], \phi^{m,0}(t) = \sqrt{2m+1}P_m(2t-1) \quad (7.1)$$

Ainsi, les premières fonctions d'échelle de la base \mathcal{M}_4 (cf. figure 7.5) seront :

$$\begin{aligned}
 \phi^{0,0}(t) &= 1 \\
 \phi^{1,0}(t) &= \sqrt{3}(2t - 1) \\
 \phi^{2,0}(t) &= \sqrt{5}(6t^2 - 6t + 1) \\
 \phi^{3,0}(t) &= \sqrt{7}(20t^3 - 30t^2 + 12t - 1)
 \end{aligned}
 \tag{7.2}$$

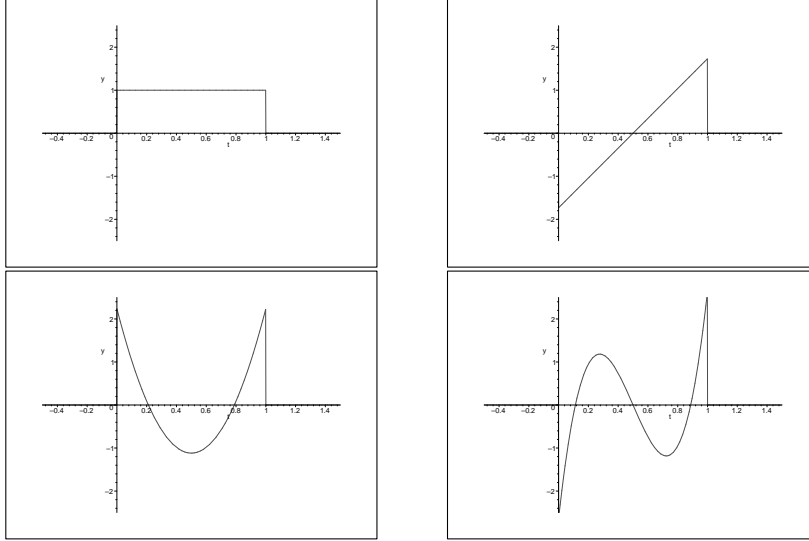


FIG. 7.5: Fonctions d'échelle de la base \mathcal{M}_4

7.2 Application à la radiosité spatiale et temporelle

Nous présentons dans ce paragraphe les modifications à apporter à l'algorithme de radiosité hiérarchique spatiale et temporelle pour pouvoir utiliser une base de multi-ondelettes. Dans un premier temps nous montrerons comment nous représentons en mémoire les radiosités et irradiances, exprimées dans une base \mathcal{M}_M . Puis nous détaillerons comment calculer les interactions lumineuses entre éléments (en particulier celles mettant en jeu un groupe, cf. paragraphe 6.2). Nous présenterons également un oracle de raffinement adapté à notre nouvelle base de fonctions. Finalement, nous terminerons ce chapitre par le calcul des expressions permettant le passage entre les différents niveaux de détail, intervenant au cours de la phase de mise à jour de la hiérarchie (cf. paragraphe 1.2.3).

7.2.1 Implémentation

Comme l'ont montré Gortler *et al.* [46], la radiosité hiérarchique est un cas particulier de la radiosité par ondelettes correspondant à la base de Haar, chaque niveau de détail correspondant à un niveau dans la hiérarchie. Dans ce cas, on constate que ne sont utilisées que les fonctions d'échelle (*i.e.* les fonctions uniformes sur chaque élément), et pas les fonctions de détails. Dans le cadre de la radiosité par ondelette, pour n'importe quelle autre base on peut procéder de même et systématiquement se passer des fonctions de détail $\psi_i^{m,j}$ en exprimant la radiosité à l'aide des seules fonctions d'échelle $\phi_i^{m,j}$ [95]. En effet, les fonctions de détail peuvent toujours être exprimées comme une combinaison linéaire des fonctions d'échelle du niveau de détail immédiatement supérieur. Nous avons fait de même et notre implémentation n'utilise que les fonctions d'échelle des bases \mathcal{M}_M .

En conséquence, la décomposition de la radiosité (vue comme une fonction du temps) dans la base d'ondelettes \mathcal{M}_M se fait *implicitement*, un niveau de détail étant ajouté pour chaque niveau de subdivision temporelle dans la hiérarchie des éléments. En pratique, la radiosité d'un élément, pour une longueur d'onde donnée, est une combinaison linéaire des fonctions d'échelle correspondant à sa profondeur dans la hiérarchie. Elle peut donc être représentée, si on utilise la base \mathcal{M}_M , par M coefficients par longueur d'onde. Pour notre part, nous nous contentons de classiques triplets (*rouge, vert, bleu*). Remarquons que pour la représentation *spatiale* de la radiosité dans le cas statique, l'utilisation des multi-ondelettes nécessite M^2 coefficients par longueur d'onde, puisque les fonctions de base sont alors formées par produit tensoriel des fonctions d'échelle de \mathcal{M}_M (cf. paragraphe 1.2.5). De plus, puisque nous connaissons pour chaque élément l'intervalle $[t^{initial}, t^{final}]$ pendant lequel il est défini, on peut calculer simplement la radiosité d'un élément à l'instant t à partir des coefficients, des bornes de l'intervalle, et de l'expression normalisée des polynômes de Legendre sur $[0, 1]$ (cf. équation 7.2). Il n'est pas nécessaire de stocker avec l'élément le niveau de détail auquel il correspond.

7.2.2 Interactions entre éléments

Discretisation de l'équation de radiosité dans \mathcal{M}_M

Reprenons la discretisation de l'équation de radiosité spatiale et temporelle telle que nous l'avons effectuée au paragraphe 3.2. Si notre maillage est constitué de L éléments, dont la radiosité est uniforme en espace et est représentée par une combinaison linéaire de fonctions d'échelle de \mathcal{M}_M en temps, alors notre base est composée de $N = M \times L$ fonctions u_i telles que :

$$u_{M \times l + m}(x, t) = \begin{cases} \phi^{m,0} \left(\frac{t - t_l^{initial}}{t_l^{final} - t_l^{initial}} \right) & \text{lorsque } (x, t) \text{ appartient à l'élément } l \\ 0 & \text{sinon} \end{cases}$$

Les $\phi^{m,0}$ étant orthonormés, les u_i forment une base orthogonale et on a :

$$\forall i, \|u_i\| = A_l (t_l^{final} - t_l^{initial}) \text{ avec } l = \lfloor \frac{i}{M} \rfloor$$

Nous avons montré au paragraphe 3.2 que l'équation de la radiosité, projetée dans l'espace généré par une base orthogonale de fonctions $(u_i)_{1 \leq i \leq N}$, nous ramenait à la résolution du système linéaire :

$$(\mathbf{I} - \mathbf{M})\mathbf{B} = \mathbf{E}$$

avec :

$$\forall (i, j) \in [1, N]^2 \quad \mathbf{M}_{i,j} = \frac{\langle \int_Y \mathcal{K}(\cdot, Y) u_j(Y) dY, u_i \rangle}{\|u_i\|^2}$$

et

$$\forall i \in [1, N] \quad E_i = \frac{\langle E, u_i \rangle}{\|u_i\|^2}$$

Rappelons que lorsque nous utilisons une base de fonctions uniformes (ce qui correspond au cas de la base de Haar pour laquelle $M = 1$), chaque coefficient de la matrice \mathbf{M} (« facteur de forme ») caractérise la géométrie de l'interaction entre deux éléments. Dans notre cas, à chaque paire d'éléments correspond désormais dans la matrice un bloc de taille $M \times M$. La notion de facteur de forme entre deux éléments k et l se généralise donc sous la forme d'une matrice d'interaction $\mathbf{I}_{k,l}$:

$$\mathbf{I}_{k,l} = \frac{1}{A_k(t_k^{final} - t_k^{initial})} \int_{T_k \cap T_l} \mathbf{G}_{k,l}(t) \int_{Q_k} \int_{Q_l} \frac{\cos\theta(p, q, t) \cos\theta'(q, p, t)}{\pi r(p, q, t)^2} v(p, q, t) dq dp dt \quad (7.3)$$

$\mathbf{G}_{k,l}(t)$ étant la matrice définie par :

$$\mathbf{G}_{k,l}(t) = \begin{pmatrix} \phi^{0,0}(\lambda_k(t))\phi^{0,0}(\lambda_l(t)) & \cdots & \phi^{0,0}(\lambda_k(t))\phi^{M-1,0}(\lambda_l(t)) \\ \vdots & \ddots & \vdots \\ \phi^{M-1,0}(\lambda_k(t))\phi^{0,0}(\lambda_l(t)) & \cdots & \phi^{M-1,0}(\lambda_k(t))\phi^{M-1,0}(\lambda_l(t)) \end{pmatrix} \quad (7.4)$$

et λ_l est le changement de variable :

$$\lambda_l(t) = \frac{t - t_l^{initial}}{t_l^{final} - t_l^{initial}}$$

On remarquera que le coefficient en haut à gauche de $\mathbf{G}_{k,l}(t)$ vaut 1 si t appartient à l'intervalle de temps correspondant à l'interaction $(T_k \cap T_l)$, et 0 sinon. Le coefficient correspondant de $\mathbf{I}_{k,l}$ est donc égal au facteur de forme $\mathcal{F}_{k,l}$ entre k et l . On notera également que les matrices d'interaction vérifient, sous une forme plus générale, les mêmes relations algébriques que les facteurs de forme (à une transposition de la matrice d'interaction près) :

$$\mathbf{I}_{k,l} = \frac{A_l(t_l^{final} - t_l^{initial})}{A_k(t_k^{final} - t_k^{initial})} {}^t\mathbf{I}_{l,k}$$

Évaluation des interactions

Le calcul de l'irradiance émise par un élément de surface se fera donc en deux temps :

1. Calcul de la matrice d'interaction : nous utilisons des règles de quadrature Gaussienne pour évaluer les intégrales de l'équation 7.3. Typiquement, nous utilisons 16 rayons (4 points sur l'émetteur et sur le récepteur) et une 6 de dates échantillons.
2. Calcul de l'irradiance : il s'agit simplement d'une multiplication matricielle $\mathbf{I}_{k,l} \times B_l$.

On notera que le coût du stockage de la matrice d'interaction varie comme le carré du nombre M de moments nuls de la base (alors que dans le cas d'une utilisation des multi-ondelettes pour la représentation spatiale de la radiosité, il varie comme la puissance quatrième de ce nombre). L'utilisation de multi-ondelettes d'ordre supérieur pour la représentation temporelle de la radiosité est donc nettement moins gourmande en mémoire que sa contrepartie spatiale.

Émission par un groupe

Dans le cas où l'émetteur est un groupe, l'interaction est gérée d'une manière similaire à celle présentée au paragraphe 6.2. Toutefois, il nous faut désormais projeter les termes angulaires ainsi que la visibilité approchée sur notre nouvelle base de fonctions. L'irradiance reçue par un élément, à l'instant t , en un point p , en provenance d'un groupe Q contenant les surfaces Q_i , est :

$$I_{recue}(p, t) = \sum_i \int_{Q_i} B_{Q_i}(q, t) k(p, q, t) v(p, q, t) dq$$

Elle sera approximée par :

$$\tilde{I}(t) = \sum_{j=0}^{M-1} \alpha_j \phi^{j,0}(\lambda_P(t))$$

avec λ_P le changement de variable permettant de ramener l'intervalle de temps du récepteur P sur $[0, 1]$. Comme les u_i forment une base orthogonale, nous pouvons exprimer les α_j sous la forme du produit scalaire suivant :

$$\begin{aligned}\alpha_j &= \frac{1}{\|u_P\|} \langle I_{recue}, \phi_j^{j,0} \circ \lambda_P \rangle \\ &= \frac{1}{A_P(t_P^{final} - t_P^{initial})} \sum_i \left[\int_{T_P \cap T_{Q_i}} \left(\int_P \int_{Q_i} k(p, q, t) v(p, q, t) dq dp \right) B_{Q_i}(t) \phi_j(\lambda_P(t)) dt \right]\end{aligned}$$

Il serait trop coûteux d'évaluer la fonction de visibilité en l'échantillonnant au niveau des surfaces (cf. paragraphe 6.2.1). Nous la factorisons donc au niveau du groupe en l'approximant par $\tilde{V}(t)$:

$$\tilde{V}(t) = \frac{1}{A_P A_Q} \int_P \int_Q v(p, q, t) dp dq$$

Ce qui donne :

$$\alpha_j = \frac{1}{A_P(t_P^{final} - t_P^{initial})} \sum_i \left[\int_{T_P \cap T_{Q_i}} \left(\int_P \int_{Q_i} k(p, q, t) dp dq \right) B_{Q_i}(t) \tilde{V}(t) \phi_j(\lambda_P(t)) dt \right]$$

Comme précédemment, ces intégrales seront calculées en utilisant une règle de quadrature Gaussienne.

Réception par un groupe

L'irradiance $I_P(t)$ reçue par un groupe P est immédiatement redistribuée à toutes les surfaces P_i qu'il contient en prenant en compte l'orientation relative de ses surfaces par rapport à la direction d'incidence de la lumière :

$$I_{P_i}(t) = I_{recue}(t) \cos \theta_i(t)$$

Nous devons reprojeter cette irradiance sur notre base de fonctions :

$$\tilde{I}_{P_i} = \sum_{j=0}^{M-1} \beta_j \phi_j(\lambda_P(t))$$

avec

$$\begin{aligned}\beta_j &= \langle I_{P_i} | \phi_j \circ \lambda_P \rangle \\ &= \frac{1}{t_P^{final} - t_P^{initial}} \int_{T_P \cap T_{Q_i}} I_{recue}(t) \cos \theta_i(t) \phi_j(\lambda_P(t)) dt \\ &= \frac{1}{t_P^{final} - t_P^{initial}} \sum_{i=1}^{M-1} \alpha_i \int_{T_P \cap T_{Q_i}} \cos \theta_i(t) \phi_j(\lambda_P(t)) \phi_i(\lambda_P(t)) dt\end{aligned}$$

Ces M intégrales seront également calculées en utilisant une règle de quadrature Gaussienne.

Nous procédons donc à deux approximations successives. Dans un premier temps, l'irradiance reçue par un groupe a été projetée sur la base de fonctions, distribuée à chacune des surfaces, puis reprojétée à nouveau sur notre base de fonctions. L'oracle de raffinement ne devrait décider de placer les liens au niveau d'un groupe que lorsque cette double approximation est suffisante. Employer une méthode de calcul des interactions plus précise mais plus lente nuirait au caractère hiérarchique de l'algorithme et donc à ses performances.

7.2.3 Un nouvel oracle de raffinement

L'oracle de raffinement que nous avons décrit au paragraphe 4.2.2 est parfaitement utilisable lorsque l'on emploie une base de multi-ondelettes. Toutefois, comme il est seulement basé sur une évaluation de la variance de la fonction noyau \mathcal{K} de l'équation de la radiosité, il ne prend pas en compte la nature de la base de fonctions sur laquelle nous avons discrétisé l'équation de radiosité. En pratique, il générera donc le même maillage quelque soit la base de fonctions employée. Or, les multi-ondelettes d'ordre supérieur nous permettent de stocker sur un seul élément plus d'information sur les variations temporelles de la radiosité que les fonctions uniformes. En conséquence, pour une qualité équivalente, elles devraient nous permettre de nous contenter d'un maillage moins dense dans la direction temporelle.

Nous présentons donc dans ce paragraphe une extension au cas de la radiosité spatiale et temporelle d'un oracle permettant de prendre en compte la nature de la base employée [11, 12, 25], basé sur une estimation de l'erreur effectuée dans la représentation de la radiosité sur le récepteur.

Principe général

Nous construisons une grille de points de contrôle situés à la surface de l'élément récepteur, et un ensemble de dates de contrôle prises dans l'intervalle de temps correspondant à l'interaction considérée. Sur cette grille, nous estimons l'irradiance en utilisant deux méthodes :

1. en calculant la matrice d'interaction, puis en évaluant l'irradiance en chaque point de la grille par interpolation. Ces valeurs d'irradiance sont celles qui seront effectivement utilisées si un lien est établi à ce niveau de la hiérarchie.
2. en intégrant directement l'équation donnant l'irradiance pour un point et une date de contrôle (p, t) en utilisant une quadrature Gaussienne :

$$I(p, t) = \int_Q k(p, q, t) v(p, q, t) B_Q(t) dq \quad (7.5)$$

La différence entre ces deux valeurs est une indication de l'erreur que l'on effectuerait si l'on établissait le lien à ce niveau. Nous utilisons la moyenne de ces différences pour déterminer si le lien doit être raffiné ou non. Nous effectuons le choix entre un raffinement en temps ou en espace en comparant deux estimations de la variance de cette erreur :

- une moyenne des variances spatiales calculées à date fixée en faisant varier la position des points de contrôle.
- une moyenne des variances temporelles calculées pour un point de contrôle donné en faisant varier la date.

Les points et dates de la grille de contrôle doivent être choisis prudemment de façon à ce qu'ils produisent le maximum d'information. En particulier, on prendra garde à les choisir différents des points et dates de quadrature utilisés pour l'évaluation des matrices d'interaction. De plus, il est conseillé de faire varier la taille de la grille de contrôle en fonction de l'étendue (spatiale et temporelle) de l'élément récepteur, de manière à ne pas manquer des événements importants au début du raffinement. De plus, afin de contrôler la continuité temporelle de la solution, il est important de placer des dates de contrôle au début et à la fin de l'intervalle de temps sur lequel est défini le récepteur, auxquels on peut attribuer un poids supplémentaire lors de l'évaluation de l'erreur.

Placement des points de contrôle dans un groupe

Il nous reste à préciser comment placer un nombre arbitraire de points de contrôle sur le récepteur si ce dernier est un groupe. Il n'est pas envisageable de placer ces points de contrôle suivant une grille fixe à l'intérieur de la boîte englobante du groupe. En effet, le calcul de l'irradiance reçue en chaque point de contrôle (cf. équation 7.5) nécessite la connaissance de l'orientation du récepteur, qui n'est pas définie si le point de contrôle ne se situe pas sur une surface.

Nous plaçons donc nos points de contrôle sur les surfaces contenues dans le groupe. S'il est composé de m surfaces, et que notre oracle de raffinement nécessite pour cette interaction n points de contrôle, nous procédons de la façon suivante :

- Si $n > m$, nous choisissons $\lceil \frac{n}{m} \rceil$ points de contrôle sur chaque surface.
- Si $n < m$, nous choisissons une surface toutes les $\lfloor \frac{m}{n} \rfloor$ et nous plaçons un point de contrôle sur chacune de ces surfaces.

De cette manière nous obtenons généralement une répartition homogène des points de contrôle dans le groupe.

7.2.4 Mise à jour de la hiérarchie

Lorsque l'on utilise une base de multi-ondelettes d'ordre supérieur à 0, le mécanisme de mise à jour de la représentation hiérarchique de la radiosité se complique quelque peu par rapport au cas des fonctions uniformes. Pour pouvoir effectuer cette mise à jour, il nous faut en effet savoir comment passer d'une représentation à un niveau de détail donné au niveau de détail immédiatement inférieur ou supérieur.

La première de ces deux opérations correspond à un filtrage de l'information (calcul de la moyenne de la radiosité des éléments fils dans le cas d'une base de fonctions uniformes), la seconde à un ré-échantillonnage (dans le cas uniforme, l'irradiance est transmise telle quelle en descendant la hiérarchie).

Calcul des coefficients de filtrage

Étant donnés deux éléments frères, l'un valide sur $[t_0, \frac{t_0+t_1}{2}]$, l'autre sur $[\frac{t_0+t_1}{2}, t_1]$, et leur élément père, défini sur $[t_0, t_1]$ dont les radiosités sont décrites respectivement par les fonctions f , g et h suivantes :

$$f = \sum_{i=0}^{M-1} \alpha_i \phi^{i,0} \circ \lambda_f \quad g = \sum_{i=0}^{M-1} \beta_i \phi^{i,0} \circ \lambda_g \quad h = \sum_{i=0}^{M-1} \gamma_i \phi^{i,0} \circ \lambda_h$$

avec $\lambda_f, \lambda_g, \lambda_h$ les changements de variables mettant en correspondance chacun des intervalles de définition avec $[0, 1]$.

Connaissant f et g , nous cherchons la fonction h appartenant au niveau de détail inférieur, approchant le mieux possible $f + g$ sur $[t_0, t_1]$. Cette fonction est le projeté orthogonal de $f + g$ sur ce niveau de détail, *i.e.* la fonction telle que $h - f - g$ est orthogonale à toutes les fonctions $(\phi^{j,0} \circ \lambda_h)_{0 \leq j \leq M-1}$. Donc :

$$\forall j \in [0, M-1], \int_{t_0}^{t_1} (h(t) - f(t) - g(t)) \phi^{j,0}(\lambda_h(t)) dt = 0$$

Or f (respectivement g) est nul en dehors de $\left[t_0, \frac{t_0 + t_1}{2} \right]$ (respectivement $\left[\frac{t_0 + t_1}{2}, t_1 \right]$), donc on a :

$$\begin{aligned} \forall j \in [0, M-1], \int_{t_0}^{t_1} \sum_{i=0}^{M-1} \gamma_i \phi^{i,0}(\lambda_h(t)) \phi^{j,0}(\lambda_h(t)) dt &= \int_{t_0}^{\frac{t_0+t_1}{2}} \sum_{i=0}^{M-1} \alpha_i \phi^{i,0}(\lambda_f(t)) \phi^{j,0}(\lambda_h(t)) dt \\ &+ \int_{\frac{t_0+t_1}{2}}^{t_1} \sum_{i=0}^{M-1} \beta_i \phi^{i,0}(\lambda_g(t)) \phi^{j,0}(\lambda_h(t)) dt \end{aligned}$$

Si on ramène les intégrales sur $[-1, 1]$, en revenant à la définition des $\phi^{i,0}$ à partir des polynômes de Legendre P_i , on obtient :

$$\begin{aligned} \forall j \in [0, M-1], \sum_{i=0}^{M-1} \gamma_i \int_{-1}^1 P_i(u) P_j(u) du &= \sum_{i=0}^{M-1} \alpha_i \int_{-1}^0 P_i(2u+1) P_j(u) du \\ &+ \sum_{i=0}^{M-1} \beta_j \int_0^1 P_i(2u-1) P_j(u) du \end{aligned}$$

Or les P_i sont orthogonaux. On sait que

$$\sqrt{i + \frac{1}{2}} \sqrt{j + \frac{1}{2}} \int_{-1}^1 P_i(u) P_j(u) du = \delta_{i,j}$$

On obtient donc les M coefficients définissant h grâce par l'équation vectorielle $\gamma = \frac{1}{2}(A\alpha + B\beta)$, A et B étant les matrices définies par :

$$A_{i,j} = 2\sqrt{i + \frac{1}{2}} \sqrt{j + \frac{1}{2}} \int_{-1}^0 P_i(2u+1) P_j(u) du \quad (7.6)$$

$$B_{i,j} = 2\sqrt{i + \frac{1}{2}} \sqrt{j + \frac{1}{2}} \int_0^1 P_i(2u-1) P_j(u) du \quad (7.7)$$

Calcul des coefficients de rééchantillonnage

Étant donné un élément de radiosité h subdivisé en temps, il s'agit ici de trouver l'expression de sa radiosité au niveau de détail immédiatement supérieur (*i.e.* celui de ses deux fils). Par des calculs similaires à ceux effectués au paragraphe précédent, on peut montrer que l'expression de la radiosité h sur les deux éléments fils est donnée par les deux équations matricielles :

$$\alpha = {}^t A \gamma \quad \text{et} \quad \beta = {}^t B \gamma$$

On notera qu'on retrouve le cas particulier de la base de Haar ($M = 1$), lorsque les matrices A et B sont réduites à un seul coefficient de valeur 1. On remarquera également que les deux matrices A et B vérifient la relation $A^t A + B^t B = 2I$, de sorte que si l'on plonge h au niveau de détail supérieur et l'on reprojette ensuite les deux jeux de coefficients obtenus au niveau courant, on obtient de nouveau la fonction h .

Exemple

A titre d'exemple, voici les matrices de filtrage pour la base \mathcal{M}_4 . Les matrices correspondantes pour les bases d'ordre M inférieur sont données par le sous-bloc principal de taille $M \times M$.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 \\ 0 & -\frac{\sqrt{3}\sqrt{5}}{4} & \frac{1}{4} & 0 \\ \frac{\sqrt{7}}{8} & \frac{\sqrt{3}\sqrt{7}}{8} & -\frac{\sqrt{5}\sqrt{7}}{8} & \frac{1}{8} \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{\sqrt{3}\sqrt{5}}{4} & \frac{1}{4} & 0 \\ -\frac{\sqrt{7}}{8} & \frac{\sqrt{3}\sqrt{7}}{8} & \frac{\sqrt{5}\sqrt{7}}{8} & \frac{1}{8} \end{pmatrix}$$

Conclusion

Les coefficients des matrices A et B étant indépendant de l'élément sur lequel on souhaite filtrer ou échantillonner, ils peuvent être pré-calculés. Le mécanisme de mise à jour de la hiérarchie résultant est donc très simple.

Algorithme 7.1 Mise à jour récursive de la hiérarchie

Mise_A_Jour(Données : *element*, *somme_irradiance_peres*, **Resultat** : *energie_totale*)

```

energie_totale ← 0
element.irradiance_recoltee ← element.irradiance_recoltee + somme_irradiance_peres
si ( Est_une_feuille(element) ) alors
    element.radiosite ← element.irradiance_recoltee × element.reflectance
    energie_totale ← element.radiosite × element.aire × element.duree
sinon
    si ( Est_subdivise_en_temps(element) ) alors
        irr_transmise ←  ${}^tA$  × element.irradiance_recoltee
        Mise_A_Jour(files_gauche(element), irr_transmise, energie_fils_gauche)
        irr_transmise ←  ${}^tB$  × element.irradiance_recoltee
        Mise_A_Jour(files_droit(element), irr_transmise, energie_fils_droit)
        energie_totale ←  $A$  × energie_fils_gauche +  $B$  × energie_fils_droit
    sinon
        pour tout  $f \in$  files(element) faire
            Mise_A_Jour( $f$ , element.irradiance_recoltee, energie_fils)
            energie_totale ← energie_totale + energie_fils
        fin pour
    fin si
    element.radiosite ← energie_totale / (element.aire × element.duree)
fin si

```

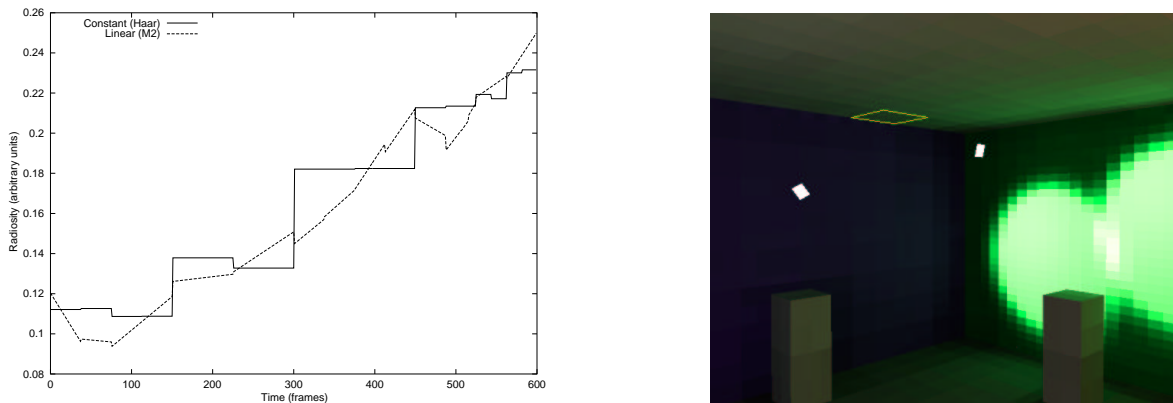



FIG. 7.6: Variations de la fonction de radiosité au centre de l'élément mis en évidence à droite (unités arbitraires)

7.3 Résultats expérimentaux

Nous avons apporté à notre programme de radiosité hiérarchique spatiale et temporelle les modifications décrites dans les paragraphes précédents afin d'utiliser la base de fonctions \mathcal{M}_2 . Cela permet à la radiosité sur chaque élément de varier linéairement.

Dans ce chapitre, nous montrons que cette nouvelle base de fonctions permet d'améliorer grandement la qualité des animations pour lesquelles l'utilisation d'éléments uniformes entraîne des discontinuités dans l'éclairage. Nous montrons également qu'elle permet une réduction du nombre de subdivisions temporelles nécessaires à l'obtention d'une précision donnée dans les échanges lumineux. Cette réduction de la hiérarchie a pour conséquence, pour ces animations, une amélioration des temps de calcul et de la place mémoire utilisée.

7.3.1 Amélioration de la continuité temporelle

Afin de démontrer l'amélioration de la continuité temporelle, nous utilisons de nouveau la scène de test employée au paragraphe 5.2.2, grâce à laquelle nous avons mis en évidence l'existence des discontinuités temporelles dues à l'utilisation d'une base de fonctions uniformes. Rappelons brièvement que cette scène est composée de quatre boîtes blanches fixes, dans une pièce illuminée par quatre projecteurs en mouvement. Ces projecteurs ne créant qu'un cône de lumière assez étroit, la quasi totalité de la scène n'est éclairée que de manière indirecte. De plus, les murs de la pièce étant respectivement rouge, vert, bleu et blanc, le mouvement des projecteurs provoque des variations importantes dans la couleur de l'éclairage ambiant. On peut en particulier observer des taches colorées se déplacer au sol et au plafond de la pièce, ainsi que sur les boîtes, provoquées par la réflexion de la lumière sur les murs.

Rappelons également que le mouvement de ces taches colorées est, dans le cas où l'on utilise une base de fonctions uniformes, particulièrement discontinu. Tout se passe en effet comme si pour ces taches de couleur, l'animation n'était mise à jour que toutes les deux ou trois secondes. On peut constater, sur le graphe de la figure 7.6 l'amplitude de ces discontinuités. On y voit clairement que, comme nous l'avons noté au paragraphe 5.2.2, les discontinuités les plus importantes se produisent au milieu de l'animation, puis aux premier et troisième quarts, puis dans une moindre mesure aux autres fractions paires de l'animation.

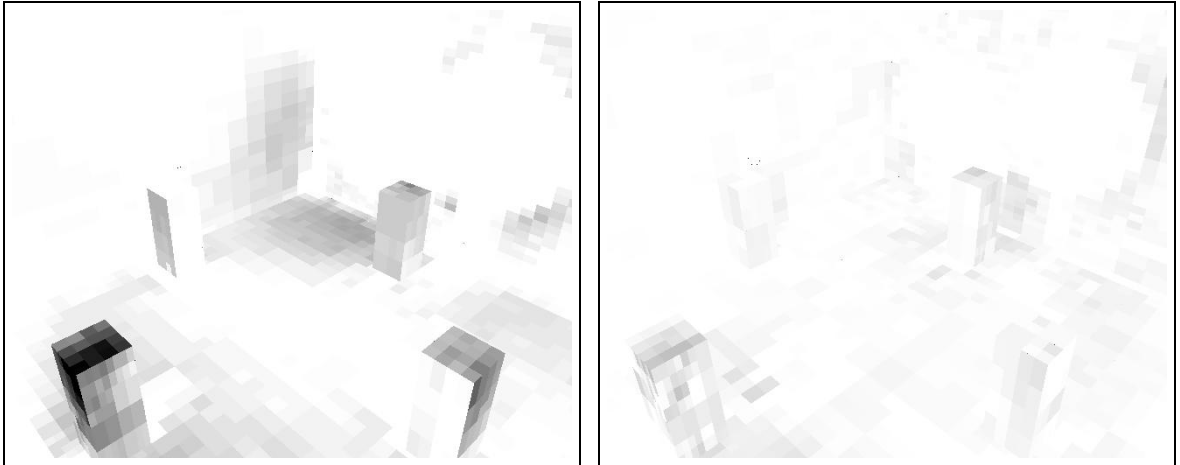


FIG. 7.7: Comparaison des discontinuités temporelles à $t = \frac{1}{2}$. Un ton plus sombre indique une plus grande discontinuité. À gauche : avec une base de Haar. À droite : avec une base \mathcal{M}_2

Si l'on calcule la même animation en utilisant une base de fonction \mathcal{M}_2 , l'éclairage indirect de l'animation obtenue est visiblement plus cohérent. A moins d'y prêter une attention particulière, il est extrêmement difficile d'y repérer quelque discontinuité. La figure 7.6 permet de constater que l'éclairage indirect est indubitablement plus « continu » que lorsque l'on utilise une base de fonctions uniformes. L'utilisation de la base de multi-ondelettes \mathcal{M}_2 ne nous permet pas de *garantir* une représentation continue de la radiosité, mais elle permet de réduire fortement l'amplitude des discontinuités.

La figure 7.7 permet de constater que la principale discontinuité temporelle, ayant lieu au milieu de l'animation, est quasiment réduite à zéro sur toute la surface de la scène, alors qu'elle est clairement visible en de nombreux endroits dans le cas de la base de Haar.

7.3.2 Performances de l'algorithme

Le tableau 7.1 donne le temps de calcul et la place mémoire nécessaires au calcul de notre animation test. À titre de comparaison, nous donnons ces chiffres pour un calcul image par image effectué à l'aide d'un algorithme de radiosité hiérarchique classique, utilisant un oracle de raffinement similaire au nôtre mais ne procédant pas à un échantillonnage temporel. Nous indiquons également les performances de notre algorithme de radiosité hiérarchique spatiale et temporelle, lorsque l'on utilise respectivement la base de Haar et la base \mathcal{M}_2 (ces chiffres sont obtenus sur un processeur MIPS R12000 cadencé à 300 Mhz).

	Temps de calcul			Mémoire
	Éclairage direct	Indirect	Total par image	
Image par Image	600 s	9.000 s	16,0 s	5 Mo
Haar	254 s	1.492 s	2,9 s	587 Mo
\mathcal{M}_2	271 s	1.172 s	2,4 s	464 Mo

TAB. 7.1: Comparatifs des performances (calcul image par image, base de Haar, \mathcal{M}_2), scène simple

Ces résultats appellent les commentaires suivants :

- On obtient un facteur d'accélération de 5.5 et 6.5 respectivement pour la base de Haar et pour la base \mathcal{M}_2 . Ceci confirme que, comme nous l'avons fait remarquer au chapitre 5, notre algorithme permet d'accélérer bien plus les scènes où l'éclairage est majoritairement indirect (on notera par ailleurs que le facteur d'accélération tombe à 2 si l'on ne considère que le temps de calcul nécessaire au calcul de l'éclairage direct). Ces chiffres sont donc bien meilleurs que ceux obtenus pour les deux premières scènes de test utilisées au chapitre 5.
- Bien que la consommation mémoire de notre algorithme reste importante (puisque nous conservons en mémoire la hiérarchie correspondant à l'ensemble des dates de l'animation), l'utilisation de la base de fonctions \mathcal{M}_2 permet de réduire la consommation mémoire d'environ 15% par rapport à la base de Haar. En dépit du sur-coût occasionné par le stockage des coefficients de radiosités supplémentaires, ainsi que des matrices d'interactions, elle permet de réduire le nombre total d'éléments ainsi que le nombre total d'interactions. Ceci permet également d'expliquer l'amélioration du temps de calcul par rapport à la base de Haar.

Afin de tester le fonctionnement de notre algorithme de regroupement de surfaces lorsque l'on utilise des fonctions d'ordre supérieur, nous avons « compliqué » la scène de test précédente, en remplaçant les boîtes par des objets plus complexes. La scène obtenue (*cf.* figure 7.9) est composée d'environ 30.000 polygones. Bien que les groupes ne soient pas raffinés très finement en temps, on peut constater sur l'animation obtenue une évolution continue de la luminosité des surfaces qu'ils contiennent conforme au déplacement des sources de lumière indirecte.

Le tableau 7.2 indique le temps et la mémoire nécessaire à ce calcul. Le facteur d'accélération obtenu est ici d'environ 14 (ce qui confirme les résultats obtenus au paragraphe 6.3). Toutefois, il semble que les approximations effectuées pour le calcul de l'illumination des groupes augmentent l'amplitude des discontinuités observées.

	Temps de calcul			Mémoire
	Éclairage direct	Indirect	Total par image	
Image par image	6s	103s	109s	17 Mo
\mathcal{M}_2	146s	4.445s	8s	415Mo

TAB. 7.2: Résultats expérimentaux, scène complexe

7.3.3 Conclusion

Nous avons, dans ce chapitre, présenté les modifications à apporter à l'algorithme de radiosités hiérarchique spatiale et temporelle afin d'employer une base de multi-ondelettes d'ordre supérieur pour la représentation temporelle de la radiosité. Nous avons montré que, puisque nous utilisons la décomposition en ondelettes sur un espace unidimensionnel (le temps), l'algorithme obtenu ne présente pas les inconvénients ordinairement associés à l'utilisation des ondelettes pour les calculs de radiosités (coût mémoire prohibitif, mécanisme de mise à jour hiérarchique compliqué, *etc.*). Nous avons également démontré que l'utilisation de la base de multi-ondelettes \mathcal{M}_2 permet de réduire de façon considérable les discontinuités temporelles qu'introduisait l'utilisation de l'algorithme de radiosités hiérarchique spatiale et temporelle. Finalement, nous avons montré que l'utilisation de ces nouvelles bases de fonctions reste compatible avec le regroupement des surfaces que nous avons présenté au chapitre 6.

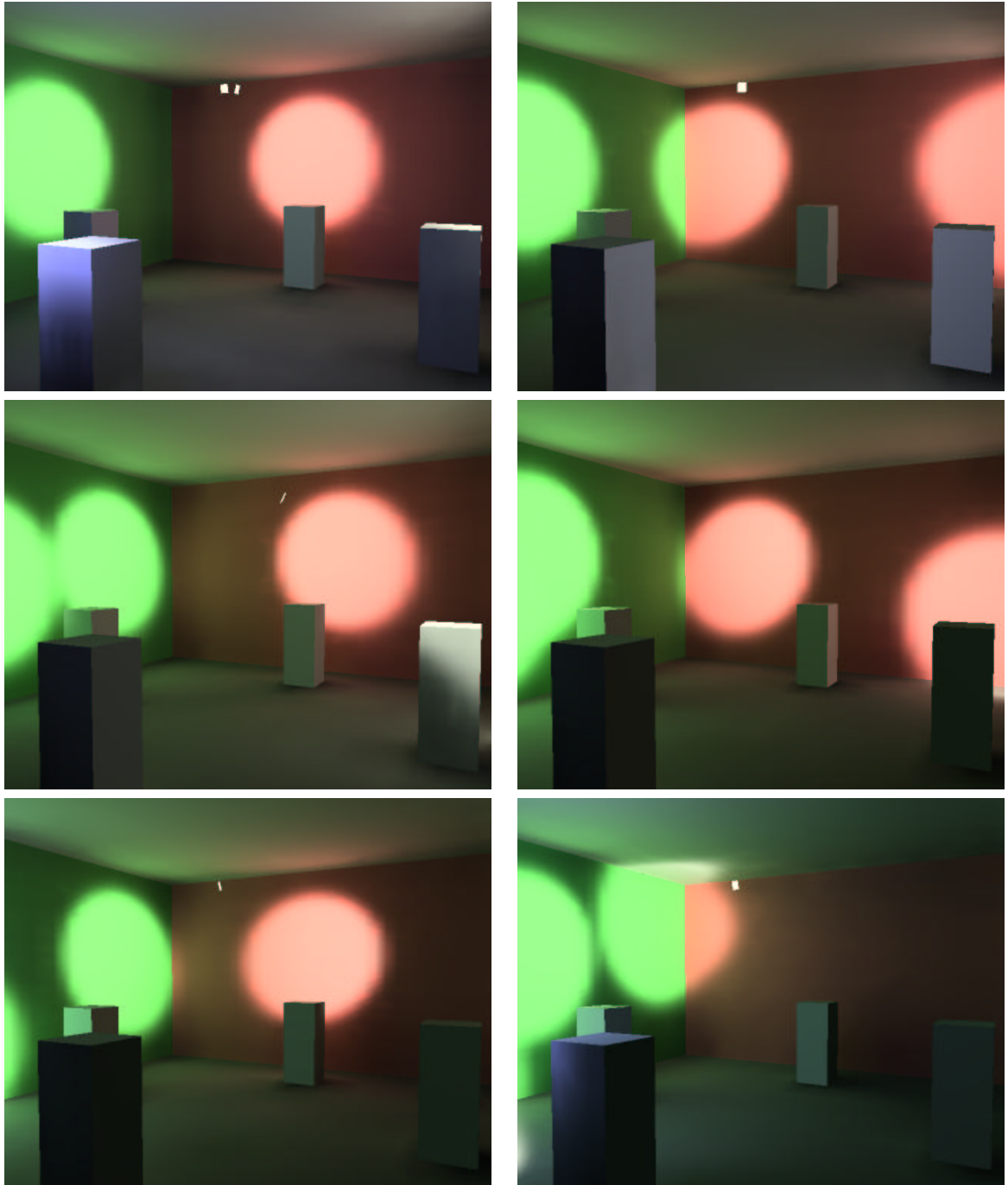


FIG. 7.8: Quelques images de notre animation testant la continuité de l'éclairage

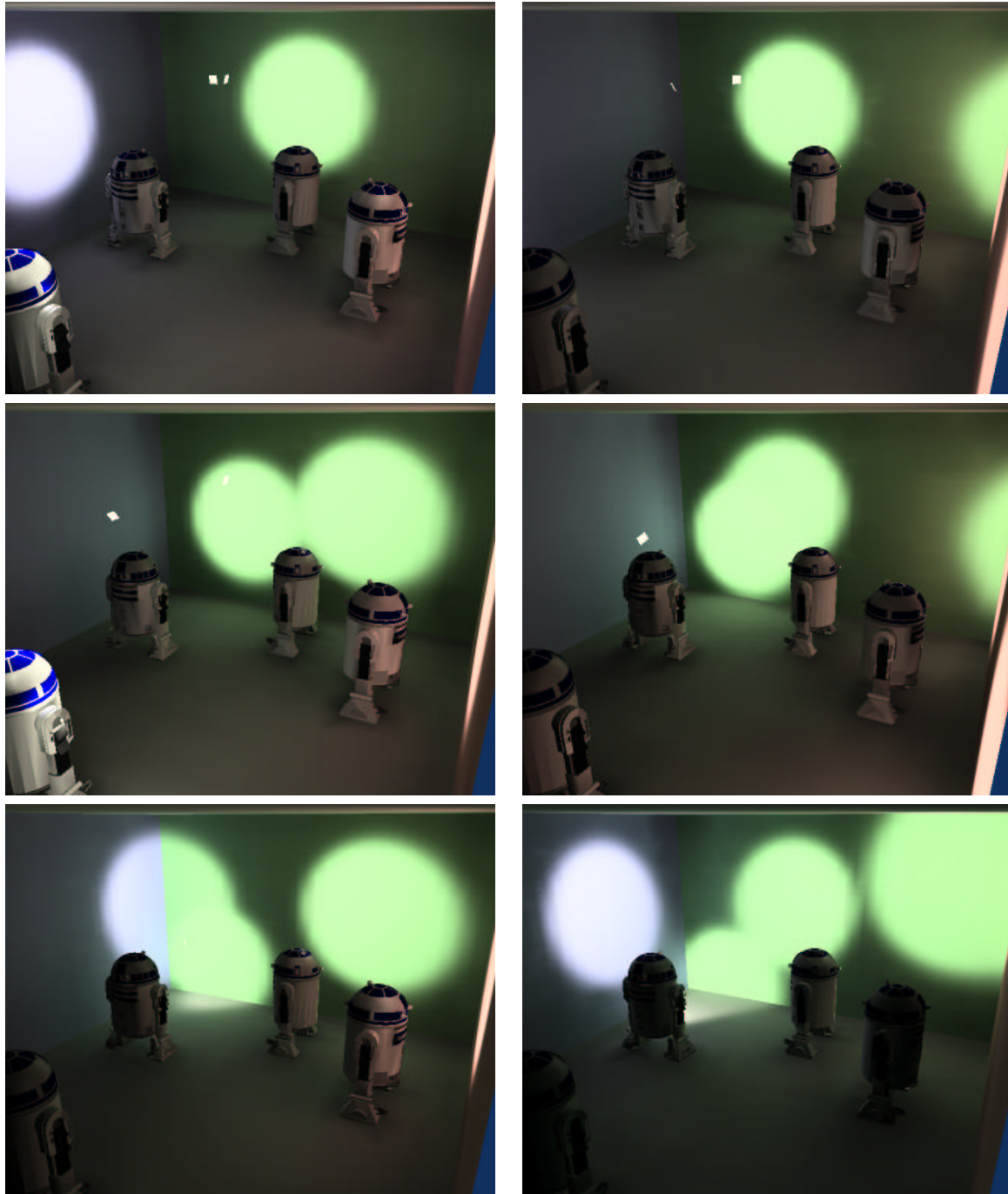


FIG. 7.9: Quelques images de notre animation testant l'utilisation combinée des groupes et des ondelettes

Gestion de la mémoire : ordonnancement des calculs

UN des inconvénients principaux de la méthode de radiosité hiérarchique quadridimensionnelle réside dans l'importance de la mémoire requise. En effet, généralisant la méthode de radiosité hiérarchique classique, elle vise à calculer une solution indépendante du point de vue et de la date. Ainsi, à tout moment de l'exécution, l'ensemble de la hiérarchie décrivant la solution calculée est présente en mémoire. Le coût mémoire en est très largement inférieur à celui nécessaire au calcul simultané de toutes les images de l'animation par un algorithme statique, mais reste toutefois trop important pour que l'on puisse envisager de calculer des plan-séquences de l'ordre de la minute sur une machine de puissance moyenne.

Ceci est d'autant plus gênant que, l'équilibre énergétique étant réalisé à chaque instant, il ne devrait pas être nécessaire de conserver simultanément en mémoire des éléments hiérarchiques correspondant à des ensembles de dates disjoints puisque ces éléments ne peuvent *physiquement* interagir. Nous sommes donc en présence d'une partition naturelle de l'espace dans lequel nous cherchons à calculer notre solution en autant de cellules « étanches » que nous avons d'images à produire. Nous présentons dans ce chapitre une série de modifications à apporter aux différentes parties de l'algorithme originel, permettant d'ordonner les calculs pour permettre de ne conserver en mémoire que des éléments pouvant potentiellement interagir.

Les travaux présentés dans ce chapitre sont en partie spéculatifs. Nous n'avons à ce jour implémenté que le mécanisme d'ordonnancement des tâches. Bien que nous n'ayons pas implémenté le mécanisme de chargement sur disque, nous pouvons calculer à tout instant la place mémoire réellement nécessaire au fonctionnement de l'algorithme. Les résultats obtenus montrent qu'il est possible de faire fonctionner l'algorithme de radiosité hiérarchique spatiale et temporelle en utilisant une quantité de mémoire largement réduite.

8.1 Principe de l'ordonnement des calculs

Dans les paragraphes suivants, nous montrons comment réduire le coût mémoire de l'algorithme de radiosit  hi rarchique spatiale et temporelle. Pour ce faire, nous ne conservons en m moire que les  l ments effectivement utiles au calcul en cours, les autres  tant temporairement sauvegard s sur une m moire de masse. Toutefois, afin de ne pas ralentir notre algorithme, il convient de r duire autant que possible la fr quence de tels acc s disque. En red finissant l'ordre dans lequel les diff rentes op rations sur la hi rarchie sont effectu es, il est possible de grouper les transferts en m moire. Cet ordre doit donc  tre bas  sur les d pendances temporelles entre les t ches ex cut es.

Puisque nous supposons que les  changes lumineux sont instantan s, l' nergie lumineuse ne circule pas   travers le temps dans notre hi rarchie. Les op rations effectu es sur la hi rarchie n'impliquent donc au pire que des  l ments dont les intervalles de temps sont inclus les uns dans les autres. En cons quence, nous regrouperons dans un m me fichier tous les  l ments et liens ayant le m me intervalle de validit .

Pour chaque type de t che, on cr era donc un agenda, structure de donn es contenant les t ches   effectuer et servant   les ex cuter dans l'ordre pr vu. Nous verrons comment ces agendas sont consult s au paragraphe 8.3. Ils permettront de d terminer quels sont les  l ments n cessaires et quels sont ceux dont la pr sence en m moire n'est plus utile, lors des diff rentes travers es de la hi rarchie. On notera qu'une approche similaire peut  tre suivie lorsque l'on souhaite parall liser de fa on efficace l'algorithme de radiosit  hi rarchique. De fait, les algorithmes pr sent s ont  t  inspir s par les travaux pr sent s dans [102] par Sillion *et al.*

Dans un premier temps, nous identifierons les diff rentes t ches atomiques que nous devons r ordonner, les parties de la hi rarchie sur lesquelles elles op rent, et les diff rentes d pendances temporelles les reliant. Nous donnerons ensuite le m canisme d'ordonnement de ces t ches se basant sur les agendas. Nous montrerons comment synchroniser la gestion des  l ments en m moire et la gestion des fichiers de sauvegarde. Nous terminerons par une discussion des r sultats obtenus.

8.2 T ches   effectuer et contraintes temporelles

Chacune des  tapes de l'algorithme de radiosit  hi rarchique (*cf.* paragraphe 1.2.3) peut  tre d compos e en t ches  l mentaires effectuant des modifications sur la hi rarchie des  l ments ou sur les liens. Habituellement, la succession de ces t ches  l mentaires prend la forme d'une simple travers e r cursive de la hi rarchie des  l ments (parcours d'arbre en profondeur d'abord). Nous souhaitons modifier l'ordre de cette travers e. Afin de d terminer quel ordre de travers e est utilisable, nous examinerons dans ce paragraphe chaque type de t che, leur d roulement et les donn es qu'elles n cessitent. Nous en d duirons les contraintes sur l'ordre dans lequel elles doivent  tre ex cut es.

8.2.1 Recensement des liens

Pour initialiser le processus de raffinement, il est n cessaire de recenser les liens pr sents dans la hi rarchie. La t che de base du recensement (*cf.* algorithme 8.1) consiste, pour un  l ment donn , pour chaque lien correspondant   une interaction pour laquelle il est r cepteur,   inscrire une t che de raffinement de ce lien dans l'agenda de raffinement. Il faut ensuite inscrire sur l'agenda de recensement les t ches correspondant aux  l ments fils.

Pour  tre ex cut e, cette t che ne n cessite donc que la pr sence en m moire de l' l ment consid r  et des liens pour lesquels il est r cepteur. La t che correspondant   un  l ment sera g n r e par la t che correspondant   son  l ment p re. Une t che de recensement donn e ne peut donc  tre ex cut e que si la t che correspondant   son  l ment p re a  t  ex cut e auparavant.

Algorithme 8.1 Tâche atomique de recensement des liens

*Tache_Recensement***Donnees :**| *identificateur_element* // cf. paragraphe 8.4.3**Execution :**| **pour** tout *lien* \in *element.liste_liens* **faire**| | **Inscrire**(*Nouvelle_Tache_Raffinement*(*lien*), *agenda_raffinement*)| **fin pour**| **pour** tout *element_fils* \in *Fils*(*element*) **faire**| | **Inscrire**(*Nouvelle_Tache_Recensement*(*element_fils*), *agenda_recensement*)| **fin pour**

8.2.2 Raffinement des interactions et collecte des irradiances

La tâche élémentaire composant le processus de raffinement (cf. algorithme 8.2) commence par l'évaluation de l'oracle de raffinement pour le lien auquel elle s'applique. Si la précision de l'échange est jugée suffisante, le lien est établi à ce niveau. L'irradiance transitant par ce lien est alors calculée. Dans le cas contraire, le lien est subdivisé (en temps ou en espace, suivant les conclusions de l'oracle) et les tâches de raffinement correspondant aux liens obtenus sont ajoutées à l'agenda.

Cette tâche nécessite donc, pour être exécutée, la présence en mémoire du lien à raffiner, ainsi que des deux éléments qu'il relie. Rappelons que l'intervalle de validité du lien est l'intersection de celui de l'émetteur et du récepteur (cf. paragraphe 4.2, équation 4.1). L'évaluation de cette tâche peut donc nécessiter la présence en mémoire d'éléments dont l'intervalle de validité contient celui du lien.

Algorithme 8.2 Tâche atomique de raffinement et de collecte d'un lien

*Tache_Raffinement***Donnees :**| *identificateur_lien***Execution :**| *avis* \leftarrow **Consulter_Oracle**(*lien*)| **si** (*Doit_Etre_Subdivise*(*avis*)) **alors**| | *liste_liens* \leftarrow **Subdivise**(*lien*, **Direction**(*avis*))

| | // On subdivise le lien soit en temps, soit en espace.

| | **pour** *nouveau_lien* \in *liste_liens* **faire**| | | **Inscrire**(*Nouvelle_Tache_Raffinement*(*nouveau_lien*), *agenda_raffinement*)| | **fin pour**| **sinon**| | **Etablir**(*lien*)| **fin si**

8.2.3 Mise à jour : Descente (*Push*)

La mise à jour hiérarchique (*cf.* paragraphes 4.1.3 et 7.2.4), nécessaire pour maintenir la cohérence entre les différents niveaux de détail, est composée de deux phases :

- Une phase de descente, où les irradiances reçues à chaque niveau de la hiérarchie sont sommées.
- Une phase de remontée, où les radiosités de chaque niveau sont filtrées pour obtenir celles du niveau supérieur.

À chaque élément de la hiérarchie correspondra donc une tâche élémentaire de descente et une tâche élémentaire de remontée.

La tâche élémentaire pour la descente commence par ajouter l'irradiance qui lui est passée en paramètre à celle de l'élément concerné. Si cet élément est une feuille, elle met à jour sa radiosité en conséquence. Sinon elle inscrit sur l'agenda de descente les tâches correspondant à chacun des éléments fils, en leur passant en paramètre l'irradiance totale. Elle inscrit également une tâche de remontée correspondant à l'élément courant.

Les tâches de descente correspondant à un intervalle de temps donné ne pourront donc être exécutées qu'après celles le contenant. De plus, comme ce sont les tâches de descente qui inscrivent sur l'agenda les tâches de remontée, ces dernières ne pourront être exécutées tant que les premières n'auront pas été complétées.

Dans le cas où l'on utilise une base d'ondelettes pour la représentation temporelle de la radiosité (*cf.* chapitre 7), il est nécessaire de calculer les coefficients correspondant à la représentation de la fonction d'irradiance au niveau de détail supérieur avant de créer les deux tâches de descente correspondant aux éléments subdivisés. De même, la tâche de remontée correspondante devra calculer, à partir des contributions des deux tâches filles, la radiosité de l'élément qu'elle doit mettre à jour (filtrage).

Algorithme 8.3 Tâche atomique de mise à jour descendante

Tache_Descente

Donnees :

```

| identificateur_element
| tache_pere
| irradiance_descendue

```

Execution :

```

| element.irradiance_recoltee ← element.irradiance_recoltee + irradiance_descendue
| si ( Est_une_feuille(element) ) alors
|   | element.radiosite ← element.irradiance_recoltee × element.reflectance
|   | energie_totale ← element.radiosite × element.aire × element.duree
| sinon
|   | tache_remontee ← Nouvelle_Tache_Remontee(element, tache_pere, radiosite_nulle)
|   | Inscrire(tache_remontee, agenda_remontee)
|   | pour tout f ∈ fils(element) faire
|   |   | irradiance ← Reechantillonnage_Irradiance(element)
|   |   | // Multiplication eventuelle par la matrice transposee de A ou de B
|   | fin pour
|   | t ← Nouvelle_Tache_Descente(f, tache_remontee, irradiance)
|   | Inscrire(t, agenda_descente)
| fin si

```

8.2.4 Mise à jour : Remontée (*Pull*)

Chacune de ces tâches est chargée de transmettre à sa tâche mère la radiosité totale que ses tâches filles lui ont transmise. De ce fait, une tâche de remontée correspondant à un élément défini sur un intervalle de temps T ne peut être exécutée qu'après celles dont l'intervalle de temps est inclus dans T .

Algorithme 8.4 Tâche atomique de mise à jour ascendante

Tache_Remontee

Donnees :

identificateur_element
tache_remontee_pere
energie_remontee

Execution :

element.radiosite \leftarrow *energie_remontee* / (*element.aire* \times *element.duree*)
Transmettre(*energie_remontee*, *tache_remontee_pere*)
// Il faut eventuellement multiplier l'energie remontee par la matrice A ou B

8.2.5 Bilan

Nous avons montré que chacune des parties d'une itération de l'algorithme de radiosité hiérarchique spatiale et temporelle consiste en une traversée de la hiérarchie des éléments ou des liens au cours de laquelle sont accomplies certaines tâches élémentaires. À chacune de ces tâches élémentaires peut être associé un intervalle de temps (celui de l'élément ou du lien auquel elle s'applique). Pour chaque tâche associée à un intervalle de temps donné, on sait qu'un grand nombre d'éléments hiérarchiques ou de liens ne sont pas nécessaires à son exécution. Nous souhaitons donc grouper l'exécution des tâches en fonction de l'intervalle de temps associé.

Nous avons vu qu'il existe pour chaque type de tâche des contraintes sur l'ordre dans lequel elles doivent être exécutées. Nous devons donc, pour chaque type de tâche, définir un ordre sur les intervalles de temps compatible avec ces contraintes. Le tableau 8.1 indique, pour chaque type de tâche, un ordre possible les vérifiant. Nous utilisons deux ordres :

$$T_1 < T_2 \iff (T_1 \subset T_2, \text{ ou bien } T_1 \text{ est chronologiquement avant } T_2) \quad (8.1)$$

$$T_1 \prec T_2 \iff (T_2 \subset T_1, \text{ ou bien } T_1 \text{ est chronologiquement avant } T_2) \quad (8.2)$$

Le tableau 8.1 indique également les relations entre un intervalle de temps T et l'intervalle de validité E des éléments (respectivement L des liens) devant être présents en mémoire pour pouvoir exécuter n'importe quelle tâche associée à T .

Type de tâche	Ordre compatible	Éléments nécessaires tels que ...	Liens nécessaires tels que ...
Recensement	<	$E = T$	/
Raffinement	<	$E \supseteq T$	$L = T$
Descente	<	$E = T$	/
Remontée	\prec	$E = T$	/

TAB. 8.1: Contraintes sur l'exécution des différentes tâches.

On notera que seules les tâches de raffinement nécessitent la présence en mémoire d'éléments correspondant à plus d'un intervalle de temps. Le raffinement est donc la partie la plus consommatrice en mémoire de notre algorithme. Il s'agit également, et de loin, de la plus longue.

8.3 Mécanisme d'ordonnement des calculs

Nous avons vu au paragraphe précédent que nous devons créer un agenda par type de tâche, dans lequel seront enregistrées les tâches à effectuer, et qui nous servira à ordonner les traversées. Nous souhaitons regrouper les tâches en fonction de l'intervalle de temps qui leur correspond. Un agenda doit donc permettre d'associer à un intervalle de temps donné un ensemble de tâches à effectuer. Nous présentons dans ce paragraphe la structure de données que nous avons implémentée pour ces agendas.

Nous créons une liste d'intervalles de temps obtenus par N subdivisions récursives de l'intervalle complet de l'animation (soit $2^{N+1} - 1$ intervalles de temps au total, dont 2^N au niveau le plus fin) à chacun desquels sera associé une pile contenant les tâches à effectuer. Par la suite, nous ferons référence à N comme étant la résolution des agendas. Ce paramètre permet d'effectuer un compromis entre le nombre de lectures et d'écritures qui auront lieu et la place mémoire nécessaire à l'exécution de l'algorithme. Lorsqu'une tâche doit être inscrite dans un agenda, elle est stockée dans la pile associée au plus petit intervalle de temps contenant (ou égal à) l'intervalle de temps qui lui correspond.

Une traversée de la hiérarchie consiste donc en un parcours, dans un certain ordre, de la liste des intervalles de temps. À chaque intervalle de temps, nous devons commander au gestionnaire de mémoire (*cf.* paragraphe suivant) de charger les éléments nécessaires, puis effectuer toutes les tâches contenues dans la pile correspondante, et finalement demander la sauvegarde des éléments dont la présence n'est plus requise.

D'après le tableau 8.1, seulement deux traversées de la hiérarchie sont nécessaires par itération. En effet, l'ordre défini par l'équation 8.1 est compatible avec les contraintes associées aux tâches de recensement, de raffinement et de descente. De plus, les données nécessaires aux tâches de recensement et de descente forment un sous-ensemble de celles nécessaires aux tâches de raffinement. Elles pourront donc être effectuées au cours d'une même traversée. Ceci permet :

- de réduire le nombre total de chargements et d'écritures sur la mémoire de masse au cours d'une itération. De trop nombreux accès au disque ralentiraient trop notre algorithme, et au final feraient perdre tout intérêt à notre approche.
- de vider l'agenda des tâches de raffinement au fur et à mesure que les tâches de recensement le remplissent, et donc de réduire la place mémoire occupée par ce dernier. En effet, celle-ci serait proportionnelle au nombre total de liens dans la hiérarchie si l'on exécutait ces deux types de tâches au cours de traversées séparées.

L'algorithme 8.5 décrit le déroulement d'une itération de l'algorithme de radiosité hiérarchique spatiale et temporelle réordonné.

8.4 Gestion des éléments et mécanisme de chargement

Le gestionnaire mémoire est chargé :

- de charger en mémoire les éléments nécessaires et de sauvegarder sur disque ceux qui ne le sont plus lorsque le mécanisme d'ordonnement des tâches le lui commande.
- de créer en temps voulu les éléments générés par le processus de raffinement.
- de fournir aux tâches les pointeurs sur les éléments présents en mémoire qu'elles souhaitent manipuler (ou de générer une erreur si ces derniers ne sont pas en mémoire).

Algorithme 8.5 Algorithme hiérarchique réordonné

```

liste ← Subdivision( intervalle_animation, granularite )
liste_1 ← Tri(liste, ordre 8.1)
liste_2 ← Tri(liste, ordre 8.2)
repete
  pour tout intervalle ∈ liste_1 faire
    Chargement_elements_raffinement(intervalle)
    pile_recensement ← Pile(agenda_recensement,intervalle)
    tant que pile_recensement ≠ 0 faire
      | tache_recensement ← Depiler(pile_recensement)
      | Executer(tache_recensement)
    fin tant que
    pile_raffinement ← Pile(agenda_raffinement,intervalle)
    tant que pile_raffinement ≠ 0 faire
      | tache_raffinement ← Depiler(pile_raffinement)
      | Executer(tache_raffinement)
    fin tant que
    pile_descente ← Pile(agenda_descente,intervalle)
    tant que pile_descente ≠ 0 faire
      | tache_descente ← Depiler(pile_descente)
      | Executer(tache_descente)
    fin tant que
    Sauvegarde_elements_raffinement(intervalle)
  fin pour
  pour tout intervalle ∈ liste_2 faire
    Chargement_elements_remontee(intervalle)
    pile_remontee ← Pile(agenda_remontee,intervalle)
    tant que pile_remontee ≠ 0 faire
      | tache_remontee ← Depiler(pile_remontee)
      | Executer(tache_remontee)
    fin tant que
    Sauvegarde_elements_remontee(intervalle)
  fin pour
jusqu'a convergence

```

8.4.1 Chargement et sauvegarde des éléments

Rappelons que le tableau 8.1 indique pour chaque type de tâche quels sont les éléments ou liens nécessaires à son exécution. Pour les deux traversées, le chargement des éléments nécessaires pour un intervalle de temps donné est simple à effectuer : il suffit de charger le fichier contenant les éléments (et, dans le cas de la première traversée, les liens) ayant pour validité l'intervalle considéré. De même, dans le cas de la deuxième traversée, tous les éléments ayant pour validité un intervalle de temps donné peuvent être sauvegardés sur disque une fois vidée la pile des tâches de remontée correspondante.

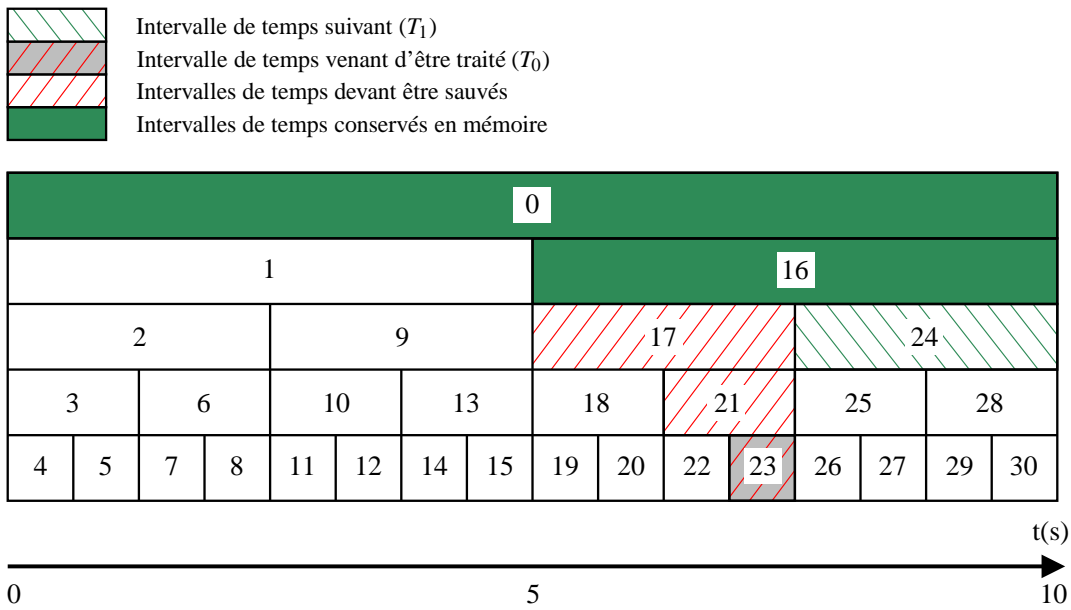


FIG. 8.1: Repérage des intervalles de temps correspondant à des éléments obsolètes lors de la traversée effectuant les tâches de raffinement. Les intervalles de temps sont numérotés dans l'ordre donné par l'équation 8.1.

En revanche, il est plus complexe de déterminer les éléments et liens qui ne sont plus utiles au processus de raffinement. Si, dans l'ordre de parcours de l'agenda, l'intervalle de temps suivant T_1 est inclus dans l'intervalle de temps actuellement traité T_0 , alors les éléments en mémoire pourront encore être utilisés par la suite. Nous n'effectuons alors aucune écriture sur disque. Dans le cas contraire, il est possible de se débarrasser de tous les éléments dont les intervalles de validité sont inclus dans l'intervalle père de T_1 (cf. figure 8.1).

Pour chaque intervalle de temps de l'agenda, il faut maintenir une liste indiquant quels sont les éléments et liens qui sont, ou qui seront, sauvés dans le fichier correspondant. Afin de réduire la place mémoire, cette liste peut également être sauvegardée avec le fichier correspondant.

8.4.2 Création des éléments

Les tâches de raffinement peuvent, par subdivision, provoquer la création de liens ou d'éléments hiérarchiques. Il n'est pas possible, dans le cas d'une subdivision temporelle, d'effectuer sur le champ cette création, puisque cela entraînerait la présence en mémoire d'éléments non souhaités. Il faut repousser cette création jusqu'au moment où le gestionnaire mémoire chargera les éléments ayant le même intervalle de temps. En revanche, il faut immédiatement leur attribuer un identificateur (cf. paragraphe suivant) pour que l'élément hiérarchique traité, et les tâches nouvellement créés puissent les référencer.

Il faut donc maintenir en mémoire une liste des éléments et liens encore à créer, pour chaque intervalle de temps de l'agenda. Cette liste contiendra l'identificateur qui leur a été attribué. Afin de pouvoir retrouver en temps utile les données nécessaires à l'initialisation de l'objet à créer, il faut également que cette liste contienne l'identificateur de l'élément ou du lien père, qui sera nécessairement encore en mémoire au moment de la création. La taille de cet « agenda de création » restera modérée puisque, de même que l'agenda de raffinement, il sera vidé au fur et à mesure de la traversée.

8.4.3 Traduction des identificateurs en pointeurs

Comme nous l'avons vu au paragraphe précédent, il est parfois nécessaire de faire référence à un objet avant même sa création. De plus, le mécanisme de chargement en mémoire et de sauvegarde sur le disque rend impossible l'attribution d'un emplacement mémoire unique à un objet. On ne peut donc, dans la hiérarchie, ou dans les tâches, faire référence à un objet en utilisant un pointeur le localisant en mémoire. On est donc obligé d'attribuer un identificateur à chaque élément (par exemple un entier) qui sera converti sur requête par le gestionnaire mémoire en pointeur. Il est donc nécessaire de construire une table de traduction, permettant d'obtenir en échange de l'identificateur d'un élément son emplacement actuel en mémoire.

De plus, il faut à tout moment pouvoir obtenir l'intervalle de validité d'un élément ou d'un lien donné (même si ce dernier n'est pas présent en mémoire). Ceci est nécessaire pour pouvoir associer un intervalle de temps aux tâches à inscrire dans les agendas. Ce tableau sera de taille proportionnelle au nombre d'éléments (respectivement de liens) présents au final dans la hiérarchie. De ce fait, si l'on emploie des agendas de résolution assez fine, il sera responsable de la plus grande partie du coût mémoire de notre algorithme (*cf.* paragraphe 8.5.1).

8.5 Conclusion

8.5.1 Résultats

Nous avons modifié notre implémentation de l'algorithme de radiosité hiérarchique spatiale et temporelle de manière à appliquer notre nouvelle politique d'ordonnancement des calculs. Nous avons implémenté un mécanisme de chargement et sauvegarde *virtuel* : tous les éléments et liens sont marqués présents ou absents en mémoire. Nous pouvons donc ainsi contrôler qu'aucun accès n'est fait à un élément qui ne devrait pas être en mémoire.

À tout instant, nous maintenons à jour un compteur des éléments dont la présence est nécessaire en mémoire. Nous évaluons également la place mémoire occupée par les structures de données nécessaires au fonctionnement de l'ordonnancement des calculs (*cf.* paragraphe 8.3) et du gestionnaire de mémoire (*cf.* paragraphe 8.4).

Afin de déterminer la quantité de mémoire que notre approche devrait permettre d'économiser, nous avons effectué des tests sur les scènes simples utilisées aux chapitres 5 (3 secondes d'animation) et 7 (24 secondes d'animation), pour différentes résolutions d'agenda. Le tableau 8.2 détaille la consommation mémoire maximum de notre algorithme, lors de la traversée correspondant aux tâches de recensement, collecte et descente (cette traversée nécessite plus d'éléments en mémoire que celle correspondant aux tâches de remontée). La figure 8.2 indique les variations de la consommation mémoire maximum en fonction de la résolution des agendas.

8.5.2 Discussion

Réduction de la mémoire

Notre approche permet de réduire, pour les scènes de test que nous avons utilisé, la consommation en mémoire vive de l'algorithme de radiosité spatiale et temporelle d'un facteur de 5 à 10. Notre algorithme gagne à être employée sur des animations de durée importante. En effet, il est basé sur un partitionnement temporel des échanges lumineux. Plus l'animation à calculer est longue, et plus nous pourrons créer de partitions différentes.

Scène	Nombre maximum		Taille maximum		
	d'éléments	de liens	des structures	de la hiérarchie	totale en mémoire
ACTEURS	34.498	145.701	8,1 Mo	101,9 Mo	20,8 Mo
ÉCRAN	38.873	169.509	9,6 Mo	120,5 Mo	24,0 Mo
PROJECTEURS	51.686	68.290	26,9 Mo	378,5 Mo	40,3 Mo

TAB. 8.2: Consommation mémoire détaillée, pour différentes scènes de test : sont indiqués le nombre maximum d'éléments et de liens présents en mémoire au cours d'une itération, la place totale maximum occupée par les structures de données du mécanisme d'ordonnancement des tâches (agenda, table des intervalles de temps, *etc.*), la taille totale de la hiérarchie et la mémoire vive maximum que devrait utiliser notre algorithme. La résolution des agendas utilisés pour ces tests est de 5.

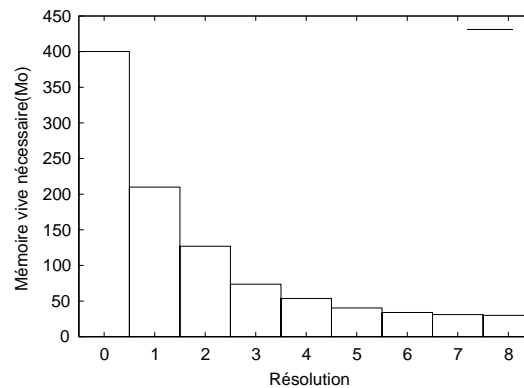


FIG. 8.2: Évolution de la consommation mémoire vive en fonction de la résolution des agendas pour la scène tests "spots". On constate que la consommation en mémoire vive atteint rapidement un plancher correspondant à la place mémoire occupée par la liste des intervalles de validité des éléments et des liens.

Le nombre d'éléments et de liens nécessaires en mémoire diminue de façon exponentielle lorsqu'on augmente la résolution des agendas. Rapidement, le coût en mémoire vive est essentiellement dû à la table de correspondance permettant d'obtenir les intervalles de temps des éléments qui ne sont pas en mémoire (*cf.* paragraphe 8.4). De plus, il est inutile, pour une animation de M images d'utiliser des agendas de résolution $N > \log_2(M) - 1$ puisqu'il n'y aura aucun élément ou lien contenu dans les fichiers correspondant à des intervalles de temps plus courts que la distance entre deux images.

En pratique, pour une animation dont la longueur est de l'ordre d'une dizaine de secondes, la consommation mémoire ne diminue donc presque plus au delà d'une résolution de 6. Si l'on souhaite réduire encore la consommation mémoire, il faudra donc, par une approche similaire à celle que nous avons suivie pour la hiérarchie, localiser quelle partie de cette table est réellement nécessaire à un moment donné de l'algorithme pour pouvoir se décharger du reste dans un fichier.

Influence sur la vitesse d'exécution

Nous n'avons pas observé de ralentissement notable de notre algorithme après implémentation du mécanisme d'ordonnancement. Rappelons toutefois que nous n'effectuons pas réellement le chargement et l'écriture des éléments et liens sur le disque. Il nous faut donc prendre en compte le temps total que devraient prendre les lectures et écritures sur la mémoire de masse si l'on souhaite conclure sur la vitesse de notre nouvel algorithme.

Les disques durs SCSI haut de gamme actuels revendiquent des taux de transferts théoriques de 160 Mo/s . En pratique, il est extrêmement difficile de soutenir un tel débit. Toutefois, notre algorithme n'effectuera que des transferts de fichiers de grande taille (écriture ou lecture en continu de l'ordre d'une dizaine de Mo). Nous devrions donc être très proche de la vitesse maximum possible en lecture ou écriture pour la mémoire de masse utilisée, si l'on utilise des agendas de résolution raisonnable. En effet, augmenter la résolution augmente le nombre d'accès disque et diminue en conséquence la taille des fichiers transférés (la quantité totale de données transférée au cours d'une itération reste inchangée).

Il faut, à chaque itération, lire et écrire l'ensemble de la hiérarchie deux fois (une pour chaque traversée). Dans le cas de la scène PROJECTEURS, pour trois itérations, cela représente environ 400 Mo , lus puis écrits 6 fois. Si l'on utilise une résolution de 5, ces 400 Mo seront répartis en $2^6 - 1 = 63$ fichiers. En pratique, on constate que le plus grand nombre d'éléments et de liens se situe dans l'intervalle de temps le plus long et dans les 32 intervalles de temps les plus courts (*cf.* paragraphe 2.2). On peut considérer que la différence de temps de transfert entre l'écriture d'un seul fichier de 400 Mo , ou d'une trentaine de fichiers de $400/32\text{ Mo}$ sera négligeable. Si l'on table sur une vitesse de transfert de 40 Mo/s , le sur-coût total dû aux transferts entre le disque et la mémoire pour le calcul de l'animation par notre algorithme serait donc de $800 \times 6/40$ secondes, soit 2 minutes. Le temps de calcul total de cette animation étant précédemment d'environ 20 minutes, l'utilisation de la mémoire de masse comme stockage temporaire ne devrait donc pas ralentir excessivement notre algorithme.

Restriction due à l'ordonnement des tâches

Notre mécanisme d'ordonnement est basé sur le fait qu'aucune tâche ne nécessite, pour être effectuée, un accès à des éléments situés plus bas dans la hiérarchie que celui auquel elle est associée. Or lors du raffinement, la répartition de l'irradiance reçue par un groupe, telle que nous l'avons présentée au paragraphe 6.2.2, viole ce principe. En effet nous distribuons immédiatement la lumière reçue aux surfaces durant la phase de collecte, sans attendre l'étape de mise à jour hiérarchique. Rappelons que cette distribution est effectuée immédiatement car elle nécessite la connaissance de la direction d'incidence de la lumière. Notre mécanisme d'ordonnement des calculs est donc incompatible avec cette méthode de prise en compte de l'orientation des surfaces.

À ce jour, nous n'avons pas implémenté le regroupement des surfaces en conjonction avec l'ordonnement des tâches. Toutefois certaines pistes pour résoudre ce problème nous paraissent prometteuses. On peut, par exemple, envisager de repousser, pour les groupes, la collecte des irradiances jusqu'au moment de la phase descendante de la mise à jour de la hiérarchie, en utilisant une méthode similaire à celle proposée par Stamminger *et al.* [110]. Au cours de cette descente une liste de liens pourra être transmise plutôt qu'une somme d'irradiances. Ces liens ne seront évalués qu'au niveau des surfaces. Puisque nous effectuons le raffinement et la phase descendante de la mise à jour de la hiérarchie au cours de la même traversée, tous les éléments de la hiérarchie nécessaires seront donc en mémoire au moment de l'évaluation du lien. Le coût mémoire dû aux listes de liens devrait rester modéré puisque l'agenda des tâches de descente est vidé au fur et à mesure que de nouvelles tâches y sont insérées.

Une fois cet obstacle levé, l'utilisation conjointe du regroupement des surfaces et du mécanisme d'ordonnement des calculs devrait être possible. Elle permettra, à condition de disposer d'une mémoire de masse suffisante, d'utiliser la méthode de radiosité hiérarchique spatiale et temporelle pour calculer des plans d'animation de l'ordre de la minute, pour des scènes formées de centaines de milliers de polygones. Au vu des résultats obtenus dans les chapitres précédents, de meilleurs facteurs d'accélération devraient alors être atteints.

Conclusion

NOUS avons dans ce mémoire, présenté un nouvel algorithme permettant le calcul de l'éclairage global diffus pour des scènes animées. Cet algorithme est basé sur la construction d'un maillage hiérarchique dans l'espace et dans le temps. Il permet ainsi de bénéficier de la connaissance *a priori* des trajectoires des objets pour « factoriser » une grande partie des calculs.

Tout d'abord, nous avons exposé les fondements mathématiques de cet algorithme. Nous avons montré que l'introduction de la variable temporelle ne changeait pas la nature de l'équation de la radiosité. Nous avons également montré que notre reformulation de cette équation pouvait être discrétisée d'une manière similaire par la méthode de Galerkin. Le système d'équations obtenues étant creux, une méthode de résolution hiérarchique apparaît particulièrement appropriée.

Nous avons ensuite montré comment l'algorithme de radiosité hiérarchique peut être étendu au cas des animations, pour produire une solution d'éclairage global diffus en tout point et pour toute date d'une animation. La principale difficulté de cette adaptation est l'établissement d'un critère de subdivision. Ce dernier permet de déterminer à quel niveau hiérarchique se dérouleront les échanges lumineux. Il revêt donc une importance capitale pour l'efficacité de la méthode comme pour la qualité de la solution produite.

L'algorithme obtenu permet d'accélérer le calcul de l'éclairage global diffus pour des scènes animées géométriquement simple. Les premiers résultats obtenus démontrent que notre approche du problème est conceptuellement valide. Notre algorithme fournit une base pour l'expérimentation de nouvelles techniques de calcul d'éclairage pour les animations. De fait, nos expérimentations nous en ayant également révélé les principales insuffisances, nous avons, pour y remédier, apporté plusieurs améliorations à l'algorithme original. Nous avons détaillé ces améliorations dans la troisième partie de ce mémoire.

En premier lieu, il était nécessaire de réduire l'ordre de grandeur du temps de calcul lorsque la complexité du maillage fourni en entrée augmente. Puisque les modèles employés dans les films d'animation sont couramment formés de centaines de milliers de polygones, un algorithme de coût quadratique est en pratique inutilisable pour ces applications. Nous avons montré que la stratégie employée dans le cas statique, et qui consiste en l'extension de la hiérarchie au dessus du niveau des surfaces sous forme de groupes, peut être ici aussi appliquée. Pour cela, nous avons présenté un algorithme paresseux effectuant ce regroupement au cours du processus de raffinement de la hiérarchie.

Nous avons également montré comment faire intervenir ces groupes dans les échanges lumineux, aussi bien en réception qu'en émission. Les résultats obtenus montrent que notre algorithme peut alors manipuler des scènes largement plus complexes, le facteur limitant étant désormais la mémoire vive requise.

Nous avons montré comment l'emploi d'une base de multi-ondelettes pour la représentation des variations temporelles de la radiosité permet de réduire les discontinuités dans l'éclairage indirect que pouvaient présenter les solutions produites par notre algorithme. Nous avons donc effectué de nouveau la discrétisation de l'équation de radiosité, en employant cette fois la base des multi-ondelettes de Alpert. Nous avons proposé un nouvel algorithme de raffinement prenant en compte le plus grand pouvoir descriptif de notre nouvelle base de fonctions, et permettant la production d'un maillage de taille moins importante. Nous avons également mis en évidence les changements induits par cette nouvelle discrétisation dans le calcul des échanges lumineux. En particulier, nous avons montré comment modifier simplement la phase de mise à jour hiérarchique pour l'adapter à notre nouvelle base de fonctions. Nous avons également détaillé les modifications à apporter aux mécanismes de calculs des échanges lumineux mettant en jeu des groupes.

Les tests que nous avons effectués prouvent que l'utilisation de cette nouvelle base de fonctions permet de réduire de façon sensible les discontinuités. De plus, lorsque l'on emploie une base de fonction d'ordre supérieur pour la représentation *spatiale* de la radiosité, on observe une augmentation importante du coût mémoire due à l'augmentation du nombre de coefficients de radiosité sur chaque élément et au stockage de la matrice d'interaction sur les liens. Dans notre cas, cette augmentation est intégralement compensée par la réduction du nombre d'éléments et de liens dans la hiérarchie due à notre nouvel oracle de raffinement.

Finalement, nous avons proposé un nouvel ordre de parcours de la hiérarchie permettant d'en transférer la majeure partie sur disque au cours d'une itération. Le mécanisme et les structures de données nécessaires n'ont été à ce jour que partiellement implémentés, mais nous ont permis de vérifier le bon comportement de l'algorithme et de mesurer, pour des scènes de test simples, le volume maximum de données nécessaires en mémoire vive pour calculer une solution de radiosité. Les résultats obtenus montrent que, sur ces scènes de test, la mémoire vive requise par notre algorithme peut ainsi être divisée par un facteur de l'ordre de 5 à 10 sans ralentissement important.

Les améliorations et résultats présentés montrent que notre algorithme ouvre des perspectives prometteuses pour le calcul rapide de l'éclairage global pour des scènes animées. Conceptuellement simple, son principe ne devrait pas poser de problème à toute personne familière avec l'algorithme de radiosité hiérarchique statique, dont il est une généralisation. Flexible, il se prête aisément à modification. Il reste cependant encore de nombreux points à explorer pour arriver à produire un algorithme permettant le rendu d'une animation de haute qualité par une méthode d'éclairage global.

Tout d'abord, en amont, comme nous l'avons rappelé dans l'introduction, se pose le problème de l'acquisition des données. Il nous faut en effet obtenir la description de la géométrie, les propriétés de réflectance des matériaux, mais également les trajectoires des points. Pour être utilisé avec notre algorithme, un logiciel permettant de générer ces trajectoires doit être en mesure de fournir des informations de haut niveau sur le mouvement des objets. Notre algorithme suppose en effet pouvoir disposer de la position et de la normale en n'importe quel point de la scène à n'importe quel date de l'animation. Il faudra donc étudier avec attention la question de la communication entre les logiciels d'animation et le moteur de rendu.

Ensuite, de nombreuses améliorations restent à apporter à notre algorithme. En premier lieu, la continuité temporelle de la radiosité, lorsque l'on utilise des groupes, peut encore être perfectionnée. Bien que celui-ci reste à développer, un oracle de raffinement plus évolué, effectuant un traitement particulier pour les groupes, devrait améliorer les résultats obtenus.

Certains points abordés dans ce document peuvent encore être approfondis. Nous n'avons pas mené à son terme le développement de l'algorithme présenté au dernier chapitre permettant la réduction du coût en mémoire vive. En conséquence, des tests en vraie grandeur restent encore à effectuer. Nous avons également laissé en suspend la question de l'opportunité d'employer des fonctions de base de degré supérieur à un. Dans la mesure où le nombre de coefficients à calculer pour chaque interaction varie dans notre cas comme le carré du nombre de moments nuls des fonctions de base, l'augmentation de ce dernier ne sera pas indéfiniment avantageuse. De plus, des problèmes de précision numérique devraient rapidement se poser.

La restriction au seul cas des surfaces purement diffuses paraît contraignante. De récents développements ont montré que l'algorithme de radiosité hiérarchique pouvait être couplé avec un algorithme de lancer de particules, afin de prendre en compte les effets directionnels [51]. Puisque notre algorithme est structurellement similaire à celui employé dans le cas statique, l'intégration de calculs stochastiques devrait ici aussi être possible. Se posera alors une fois de plus la question de la continuité temporelle de l'animation. Une approche inspirée de celle de [78] devrait réduire le bruit généré par une telle méthode.

En aval de notre algorithme, il faudra également se poser la question de la reconstruction d'images de qualité à partir de la solution d'éclairage global. Comme nous l'avons fait remarquer dans notre état de l'art, les méthodes de récolte finale actuellement employées nécessitent un temps de calcul jusqu'à dix fois supérieur à celui de la production de la solution de radiosité. Des travaux récents menés dans ce domaine ont montré que la cohérence temporelle pouvait ici aussi être mise à profit pour réduire les temps de production [72, 73]. Il faudra déterminer dans quelle mesure de telles approches peuvent être couplées avec la nôtre et bénéficier des calculs déjà effectués.

Les résultats des travaux présentés dans ce mémoire sont encourageants, mais ne représentent donc qu'un premier pas vers la production d'un algorithme intégré permettant la synthèse d'une séquence animée bénéficiant d'une simulation de l'éclairage global. Nous espérons qu'ils contribueront à enrichir la palette des algorithmes à disposition des studios d'animation.

Etude des algorithmes de regroupement pour la radiosité hiérarchique

NOUS incluons dans cette annexe l'article "*A Practical Analysis of Clustering Strategies for Hierarchical Radiosity*"[57], précédé d'un bref résumé en français, afin de compléter l'argumentaire du paragraphe 6.1. L'étude menée pour la rédaction de cet article nous a en effet permis de mieux appréhender le fonctionnement, les avantages et les inconvénients des différents algorithmes de regroupement que nous y recensons.

Analyse pratique des stratégies de regroupement pour la radiosité hiérarchique

Cet article propose différents critères d'évaluation des algorithmes de regroupement de surfaces pour la radiosité hiérarchique, dans le cadre d'une utilisation pratique, sur des modèles de provenance industrielle de taille importante, et présente une étude comparative des différents algorithmes existants ainsi que de deux algorithmes inédits. Dans un premier temps, nous y proposons une classification des stratégies de regroupement connues en fonction des structures de données sur lesquelles elles s'appuient et de la méthode de construction employée.

On peut distinguer d'une part les algorithmes produisant des groupes sous la forme d'une structure régulière (par exemple un Kd-arbre ou un *octree*), et d'autre part ceux construisant une hiérarchie de boîtes englobantes. Les premiers sont généralement plus simples à mettre en œuvre que les seconds, mais souffrent de schémas de construction trop rigides, et éprouvent notamment des difficultés à positionner un objet dans la hiérarchie à un niveau correspondant réellement à sa taille.

On peut également identifier deux familles d'algorithmes de construction d'une hiérarchie de groupes. Il est possible d'obtenir une telle hiérarchie de façon automatique en insérant les objets les uns après les autres dans la hiérarchie en partant de la racine (approche descendante ou *top-down*), ou bien en regroupant progressivement surfaces et groupes de surfaces (approche ascendante ou *bottom-up*). La principale conséquence du choix de l'un ou l'autre procédé de construction porte sur la vitesse de construction de cette hiérarchie. En effet, les approches ascendantes s'apparentent généralement à un processus d'optimisation et sont donc beaucoup plus lentes que les approches descendantes.

Nous identifions ensuite les propriétés que l'on juge désirable chez un algorithme de regroupement. Il est, en particulier, extrêmement important que la hiérarchie de groupe construite n'entraîne pas un comportement imprévisible du processus de raffinement, afin d'offrir un compromis intuitif entre temps de calcul et qualité de la solution produite. D'autres critères d'évaluation, comme la présence de chevauchements entre groupes, ou la fidélité de ces derniers à la forme des objets qu'ils contiennent, sont pris en compte.

Puis nous procédons à la description d'un protocole expérimental permettant l'évaluation des différentes stratégies étudiées. Nous avons utilisé un jeu de scènes de test industrielles géométriquement complexes sur lesquels nous avons cherché à évaluer la qualité de la hiérarchie de groupes produite, son influence sur l'utilisabilité du processus de raffinement et sur la qualité de la solution produite. Nous avons soumis à ces tests les algorithmes suivants :

PROXI : Regroupement par proximité (ascendant)

OKDT : Kd-arbre avec superposition (descendant)

OKDT-P : Kd-arbre avec facteur de branchement limité(mixte)

TF-OCT : Boîte englobantes à partir d'un *octree* (descendant)

Les résultats obtenus nous permettent de tirer un certain nombre de conclusions. En premier lieu, il apparaît que pour les scènes les plus complexes, l'évolution du temps de calcul et de la qualité des images en fonction de la tolérance à l'erreur fixée par l'utilisateur n'est ni régulière, ni monotone. L'amplitude de ce phénomène est particulièrement importante pour les scènes composées de nombreux polygones de petite taille. Le comportement le plus régulier est généralement obtenu, pour notre jeu de scènes de test, pour l'algorithme PROXI, qui malheureusement est celui qui donne les résultats les moins précis pour un temps de calcul donné.

En ce qui concerne l'utilisation de la hiérarchie de groupes en tant que structure d'accélération pour les requêtes de visibilité, il ressort de notre étude que le facteur de branchement moyen de la hiérarchie a une influence directe sur ces performances. L'algorithme que nous avons baptisé OKDT-P fournit, de ce point de vue, généralement les meilleurs résultats.

Il ressort principalement de cette étude qu'aucune méthode de regroupement n'apparaît clairement comme étant « la meilleure ». En revanche, nous avons pu identifier certains points forts et défauts pour chacune d'elles. Il apparaît également que les algorithmes de regroupement disponibles à ce jour se comportent de façon très satisfaisante sur une large catégorie de scènes de test. En revanche, leur performances sont décevantes sur des scènes composées de nombreux petits polygones. Pour ces dernières, il est nécessaire de développer une stratégie de regroupement adaptée, par exemple permettant de prendre en compte la morphologie des objets.

A.1 Introduction and Motivation

In scenes with great geometric complexity containing hundreds of thousands or even millions of polygons global illumination algorithms require the grouping, or clustering, of the individual primitives. In this way light exchanges can be treated at the level of the clusters and thus the computational complexity of the radiosity solution becomes manageable [105, 100]. Unfortunately, approximations made by hierarchical radiosity algorithms using clustering are very sensitive to the quality of the cluster hierarchy. Due to the complexity of the algorithms and data structures (by definition we are working with very complex models), no experimental analysis of the behavior of clustering algorithms has been undertaken today. Willmott and Heckbert [120] provided an inspiring study for progressive refinement radiosity and hierarchical radiosity without clustering, but evidently this study was restricted in the type of scene considered.

In this practice-and-experience paper, we investigate clustering algorithms for hierarchical radiosity in a practical context. In particular, we have chosen an experimental approach, by comparing the performance of different clustering algorithms. We have concentrated our attention on models provided by real-world applications, in an attempt to uncover problems which real users of radiosity will encounter.

We propose a taxonomy of clustering algorithms, based both on the type of data structure used, and the type of construction algorithm. We then proceed to define requirements for a clustering algorithm. In particular, a clustering algorithm should provide the user with an intuitive time-quality tradeoff : the more time is spent on a solution, the better the quality of the solution. Several other desirable properties are also identified, such as limiting the overlap of clusters, optimizing the “tightness” of the fit of clusters around objects and appropriate size of the clusters with respect to the contained objects.

Once the requirements have been defined, we proceed with a series of experiments run on models used mainly in real-world applications. Various parameters are measured, including the image quality for varying simulation parameters, the cluster construction time, the quality of the hierarchy using different criteria and the speed of ray-tracing for each cluster hierarchy.

The results of these experiments have allowed us to observe a number of interesting properties of clustering, which are discussed in detail. This in-depth study of clustering leads to the understanding that there exists no universal, ideal clustering method, while explaining the relative merits of various approaches.

A.2 A Taxonomy of Clustering Algorithms

Clustering for hierarchical radiosity was introduced by Smits *et al.* [105] and Sillion [99, 100]. Clustering algorithms can be classified by considering two aspects important to their usage :

1. The choice of data structure used. Broadly speaking, two categories have been presented : regular, typically axis-aligned subdivisions of space and hierarchies of bounding volumes (HBV), which are more closely adapted to the object geometry. Examples of such structures include k -d trees and Octrees. Hybrids have also been proposed but have not been used to date in clustering for radiosity.
2. The type of construction algorithm. Again, two basic categories have been used : top-down and bottom-up construction.

A.2.1 Data Structure Choices

The data structures used for clustering have been mostly inherited from traditional spatial subdivision structures used in graphics.

Octrees and k -d Trees

Regular structures such as octrees or k -d trees have several advantages :

- They are fast to build (see Section A.5.2), and easy to construct since the form of the clusters is (nearly always) predefined.
- They can provide fast ray-tracing since they use traditional ray-traversal mechanisms (see Section A.5.2).

Their disadvantages are not specific to clustering for illumination, but due to the rather inflexible nature of their construction :

- Objects which intersect cell boundaries do not have a trivial placement in the tree. As a consequence a heuristic needs to be determined to place the object at an appropriate level.
- The tree can be very deep if the scene contains objects with large differences in scale.
- Since the shape of sub-clusters is prescribed by the subdivision mechanism, many empty clusters can be created. These clusters consume memory and resources since they are considered in all radiosity operations.
- Cluster boundaries do not tightly fit the set of contained objects, resulting in poor-quality estimates of the optical density, for the volumetric estimation of visibility [100].

The first clustering algorithm to use k -d trees for illumination was presented by Sillion [99]. This approach uses a traditional, axis-aligned k -d tree into which objects are inserted. The strategy chosen for placement of objects intersecting cell boundaries is to put objects at the lowest level entirely containing them. For many models, this can have very negative consequences since a large number of objects can end up at very high levels in the hierarchy, with adverse effects on computation speed. Since the refinement algorithm must, when refining a link to a cluster, create new links for each of its children, a high branching factor may result in long computational times.

Hierarchy of Bounding Volumes

Algorithms based on bounding volumes hierarchies have been used by several researchers [105, 100, 38]. These algorithms use rectangular axis-aligned bounding boxes, and share the following qualities :

Advantages :

- If built correctly, the cluster hierarchy adapts well to the organization of the scene into individual objects (possibly each having its own sub-cluster hierarchy).
- An “intuitive” hierarchy can be produced for a scene with very different object sizes, without creating empty clusters.
- Clusters can be made to tightly fit their contents.

Disadvantages :

- Overlapping clusters are generally unavoidable.
- Bad clusters often result when the scene is considered as a set of individual polygons, without taking advantage of the object structure (*e.g.*, clusters mixing parts of different nearby objects).

Hybrid data structures have also been developed for clustering objects in different domains (notably for ray-tracing acceleration). Cazals *et al.* [17, 18] construct hierarchies of uniform grids, and Klimazewski *et al.* [67] present a similar approach. These methods are however designed to optimize

ray-tracing by constructing regular grid structures, and are thus unsuitable “as is” for clustering. Some ideas however, in particular those concerning grouping of objects, by Cazals *et al.*, could be applied in part to future clustering algorithms.

A.2.2 Construction Algorithms

In this section we re-visit the algorithms described above by construction algorithm type. The basic approaches are top-down and bottom-up construction. We will briefly discuss some issues of manual clustering, which is often used in industry and even in research.

Top-down Clustering

Typical top-down construction algorithms include the k -d tree (KDT) construction used by Sillion [100]. An initial cell is created, and objects are subsequently added into the cell by appropriately subdividing the cell so that the object “fits” in a sub-cell. As mentioned above, objects crossing cell boundaries are placed high up in the hierarchy.

Christensen *et al.* [21] build a hierarchy of bounding volumes starting with the bounding box of the entire scene. The bounding box is split into eight octants. For each surface contained in this bounding box, if the size of the object is smaller than that of an octant, the object is inserted into the octant containing its centroid. Otherwise, the object is attached as a direct child of the cluster at this level. A new bounding box of each octant is computed, and the algorithm continues recursively in the same manner. In this paper we refer to this algorithm and data structure as TF-OCT (“tight-fitting octree”).

Bottom-up Clustering

Bottom-up construction of clusters is inherently more complex, since it requires the examination of the existing objects and their mutual spatial relationships. Since it is in a certain sense an optimization process, the complexities of algorithms suited to such constructions rapidly become quadratic or higher in the number of objects to be processed.

Smits *et al.* [105] mention they use a “modified Goldsmith-Salmon algorithm” without describing the specifics. Sillion and Drettakis employ a hierarchy of regular grids to filter the objects and clusters in order of increasing size, and produce candidate clusters based on spatial proximity [101]. The same algorithm is used by Gibson and Hubbold [38]. In the rest of this paper we refer to this method as PROXI, for “Proximity clustering”.

Note that as we choose to group objects based on a minimization function (*e.g.*, minimize the volume of the clusters created with respect to the objects being inserted) [101, 38], an optimal solution requires an exhaustive test of all the combinations of groupings of the objects being considered. Clearly, this expense is extremely costly. A more appropriate grouping approach could be that of Cazals *et al.*[18].

Manual Construction

The difficulty of clustering is such that automatic methods are not able to treat all scenes effectively. As a consequence user intervention is inevitable at some stage in the process. Examples of such intervention are the definition of “natural clusters” such as those defined by the group of polygons belonging to a single “object” (a chair for example) or a logical group such as the set of all objects on top of a desk. The special case of touching objects is also important, since it is a way of defining object hierarchies.

Some of these interventions can be handled at input (often the set of objects defining a chair object is defined as a group by the modeling program and then instanced). Such information should be incorporated by the clustering system and used to its advantage when available. Other cases are much harder (*e.g.*, the “objects on the desk” case, or touching objects).

A.2.3 Improved Approaches

An extended version of KDT, which we call OBT for “Overlapping Binary Tree”, was derived as an attempt to merge some of the benefits of Christensen’s octree construction and the binary trees obtained with KDT. The algorithm is very similar to the KDT construction, but here, objects crossing the cell boundaries are pushed down in the hierarchy only if the ratio of their size and the considered cell size, is higher than an overlap parameter which can be set by the user (when set to 0, a KDT hierarchy is obtained). The clusters produced by this algorithm can be larger than those of KDT, and therefore may overlap, but the parameter provided gives us control on the maximum potential overlap between adjacent clusters. Specifically, for a given value of the overlap parameter λ , the following relations are true :

- maximal size of the KDT cell to which an object of size S is assigned :

$$x = \frac{2S}{\lambda}$$

- maximal overlap between two adjacent OBT clusters of original size d :

$$o = \frac{8}{3}\lambda d$$

These relations ensure that large clusters will not be overwhelmed by large (and spatially sparse) collections of small objects.

As a consequence, the OBT algorithm produces a deeper hierarchy than KDT, and thus even more empty clusters. In order to get rid of empty clusters, we developed another algorithm that uses an auxiliary OBT hierarchy to build HBV clusters in a second pass. In this second step, we keep only the bounding box of the contents (objects and child clusters) of each non-empty OBT cluster. Therefore, each cluster in the obtained HBV hierarchy will have at most two child clusters. We call the resulting new structure and algorithm OKDT (Overlapping KDT).

Unfortunately, OKDT proved to be unable to separate objects formed of thin, long polygons, such as cylinders, because they were unlikely to be inserted deep in the hierarchy. As a consequence, several clusters contain too many surfaces. This had adverse consequences on ray-casting and refinement time (average branching factor being central for hierarchical algorithms). Therefore, in order to improve our algorithm we also tried applying a PROXI clusterisation process on each cluster containing more than a dozen polygons. We call this algorithm OKDT-P (Overlapping KDT with Proximity second pass).

A.3 Requirements for a Clustering Algorithm

The most important goal for a good clustering algorithm is to group the objects in a way which represents light transfer to or from the group in the most faithful manner. At the heart of the hierarchical radiosity algorithm is the assumption that it is possible to replace a “complete” calculation by a simpler one, performed using simplified representations such as clusters. This idea can only be exploited if the resulting simplification only introduces modest perturbations in the calculation.

Unfortunately, a precise definition, or even a set of quantitative yardsticks allowing us to evaluate the quality of such a representation do not currently exist. Instead, a set of heuristics have been developed by various researchers in this domain (*e.g.* [21, 101, 38]). Considering the existing body of work, we can identify two sets of desirable properties for a clustering technique : those affecting the quality of the simulation, and those affecting the overall efficiency of the applications.

A.3.1 Requirements Regarding the Quality of the Results

We outline below some of the required properties in order to arrive at a satisfactory simulation of light transfer :

1. Monotonicity with respect to light transfer precision. If a light transfer previously represented at a certain level becomes represented at a finer level of the cluster hierarchy, the precision of the light transfer should be increased. If we consider the Time-error graph obtained by plotting the computation time as a function of the solution error, for different tolerance thresholds, we would like to obtain a *smooth and monotonic* function.
2. Overlapping clusters should be avoided as much as possible. Overlapping is problematic mainly because it implies the treatment of the transfer of light of a volume to itself, which is difficult to represent and to express in terms of the hierarchical radiosity formalism. This is especially true in the case where error bounds are estimated to drive the hierarchical refinement.
3. The nature of object group shapes should be preserved as much as possible. Clusters which contain large regions of empty space and scattered small objects should be avoided. Although this may seem evident, many automatic clustering algorithms have trouble respecting this requirement.
4. Objects should always belong to a cluster of "appropriate" size, with respect to their own dimensions. This requirement is difficult to quantify, but is especially important in the context of approximate visibility calculation, where the attenuation of light passing through clusters is estimated based on a volumetric analogy. This analogy relies on the calculation of an optical density for each cluster, which is only meaningful for clusters with well-distributed (*i.e.* "random") collections of similarly-sized objects [99].

A.3.2 Efficiency Considerations

Considering the major impact of cost considerations on the usability of clustering radiosity systems, particular attention must be paid to the two following aspects :

Building the Hierarchy

First, we are of course concerned about the efficiency of hierarchy construction. Even though the actual clustering phase is generally a preprocess, and can sometimes be stored with the model, it is still preferable to have efficient construction algorithms :

- Fast cluster construction is important in the modelling/design stage where the model changes significantly and thus long clustering times hinder lighting experimentation.
- For extremely large models it may be impractical to store an additional high-overhead data structure describing the cluster hierarchy.
- Finally, the application of hierarchical radiosity with clustering to dynamically changing environments [30] may require at least a partial rebuild of the cluster hierarchy at interactive rates.

Acceleration of Ray Casting

Another important problem is the potential use of the cluster hierarchy as a supporting data structure to accelerate ray casting queries. Such queries are used for image generation or more often, in the case of radiosity, for visibility calculations when computing form-factors. Indeed, ray casting is the method of choice for visibility estimation in hierarchical radiosity, because the hierarchical algorithm fragments the calculation into a large number of individual queries, each relative to a different emitter/receiver pair [55]. Global approaches such as hemi-cube calculations are therefore inappropriate for hierarchical radiosity.

The requirements of ray-tracing acceleration are often contradictory with those of clustering for illumination. For example, structures which minimise the number of intersections on average in a statistical sense, such as the algorithm of Goldsmith and Salmon [42], result in clusters which contain elongated bounding boxes containing large empty spaces. These clusters are inappropriate for light transfer estimation or optical density estimation, as outlined above.

A possible solution is the creation of two separate structures, one for clustering and one for ray-tracing acceleration. This however would be undoubtedly far too expensive in memory for very large models. In practice, we have observed that the performance cost implied by the use of the clustering structure for ray acceleration is most often acceptable. Some comparative results concerning the performance of various cluster hierarchies as ray tracing accelerators are presented in Section A.5.

A.4 Experimental methodology for evaluation clustering algorithms

A.4.1 Methodology

As stated earlier, clustering is necessary to make the illumination computation of industrial scenes tractable. Its most important drawback is that it makes error control very difficult, where it would be necessary to allow fast solutions to be calculated with an acceptable precision. Given the previously described clustering algorithms, we need to determine the critical parameters for their behavior.

Until now, most of the scene models used in the literature were designed by researchers for research purpose. We feel that performing our study on such scenes would have in some way "hidden" most of the problems involved by clustering. In raw "industrial" scenes, poor quality initial meshing, dense distribution of objects or randomly ordered polygons can make clustering algorithms barely usable.

Thus, our approach has been to perform a sufficient number of experiments on a set of complex, "real life" scenes in order to understand and compare each algorithms behavior in terms of quality-versus-time tuning. We limited the range of our experiments to fairly coarse and fast calculations where the impact of clustering is significant.

A.4.2 Clustering Strategies Studied

The following table summarizes the set of clustering strategies considered in this paper.

PROXI : Proximity cluster. Bottom-up construction after size filtering.

OKDT : Overlapping k -d tree. Top-down allowing partial overlap.

OKDT-P : Overlapping k -d tree with limited branching. OKDT modified to re-cluster cells with many children, using PROXI locally.

TF-OCT : Tight-fitting octree. Top-down, layer-by-layer octree construction, with re-fitting of octree cell before subdivision.

We did not submit the KDT algorithm to our tests because our experience with it proved that its average branching factor was far too high for it to be usable with scenes as large as the one we used.

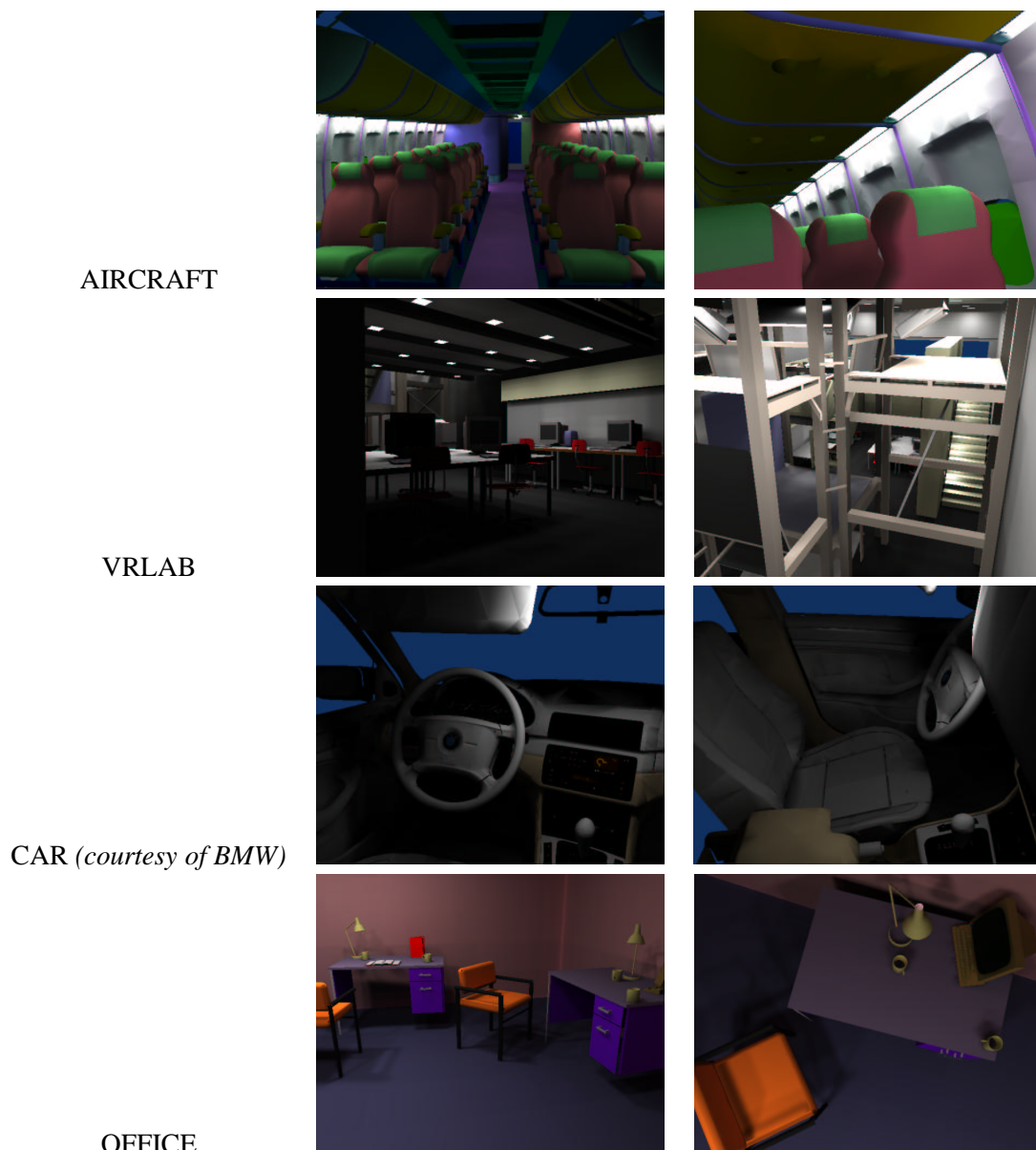


FIG. A.1: The four test scenes.

A.4.3 Test Scenes Chosen

We performed our tests on four different scenes, shown in Figure A.1. Three of them are rather large industrial-type models while the fourth one is provided as a comparison to show that clustering usually behaves very well when applied to small scenes designed for research purposes. We have taken these scenes as representative scenes for the clustering problem, allowing us to identify the different problems of the algorithms which we will test.

AIRCRAFT (184,456 polygons)

Model of an aircraft cabin (courtesy of LightWork Design Ltd). All objects have been tessellated into (rather small) triangles to account for the rounded shapes.

VRLAB (30,449 polygons)

A virtual reality lab with two floors and mostly overhead lighting (courtesy of Fraunhofer Institut für Graphische Datenverarbeitung). This scene has a mixture of large polygons, likely to be subdivided, and very small patches (on chairs and desktop computers).

CAR (216,157 polygons)

A model of a car interior with very small details, lit by a single overhead console fixture (courtesy of BMW).

OFFICE (5,260 polygons)

A model of a simple office scene. This model is much smaller than the other three and is provided to show that clustering performs well on this kind of small scene usually found in the literature.

A.4.4 Tests Performed

For each of these scenes, and for each clusterizer we decided to run the following experiments, designed to measure key aspects of clustering :

- **Data structure quality** : We listed the number of clusters in the hierarchy, as well as the average number of surfaces and clusters in each cluster node. These figures are needed to estimate both the memory cost of the hierarchy (compared to the cost of the actual geometry of the scene) and its efficiency for hierarchical radiosity. Recall that the computation speed of cluster based hierarchical radiosity depends on the average branching factor because the refinement algorithm must, when refining a link to a cluster, create new links for each child of this cluster.
- **Measure of time needed to build the cluster hierarchy.**
- **Solution quality** : To evaluate image quality we chose to use a "visual quality" error evaluation on images obtained for a given set of viewpoints rather than a view-independent error metric since we want to study the visual quality rather than the accuracy of the energy transfer quantities. Images were compared to reference images (resulting from a maximum precision calculation) using the following metric : we transform all pixels from RGB space into chromaticity space XYZ. The global image error is then the L_2 norm of the pixel-by-pixel difference image between the current and the reference image, evaluated in CIELUV space [31].
- **Algorithm usability** : To study the "quality versus time" behavior of each algorithm, we rendered each scene using each clustering algorithm, changing only the parameters controlling our refinement process (we used a BF-like refinement algorithm [55], and an error-bound based refinement, both of them showed similar behavior on clusters), measuring the time needed and evaluating the resulting image quality using the method previously described. Since we are interested in the effects of the cluster hierarchy, we do not perform tests on very precise solutions, which involve only surface-to-surface energy exchanges. Instead, we chose our refinement pa-

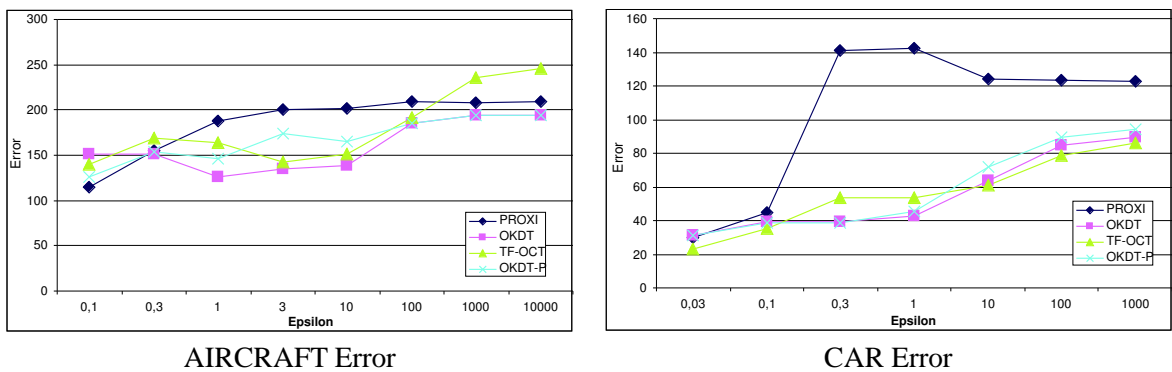


FIG. A.2: Error curves as function of user-supplied tolerance

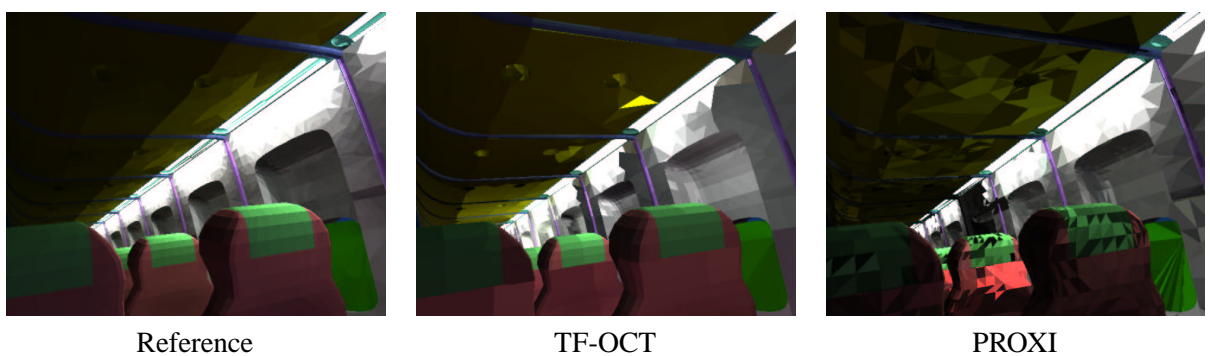


FIG. A.3: Example solutions exhibiting different (visual) forms of error.

rameters in such a way that the proportion of energy gathered through links involving clusters is significant (most of the time varying between 30% and 100%).

- **Ray casting acceleration** : As we said in Section 3.2.2, we have seen that it is desirable to use the cluster hierarchy as an acceleration structure to answer visibility requests (see Section A.3.2). We decided to test the efficiency of each structure for ray-casting. Therefore, we chose 100,000 random pairs of surfaces in each scene, and measured the total time needed to search for a possible occlusion between each pair.

We also ran a separate set of experiments to evaluate the efficiency of the cluster hierarchy when using approximated volumetric visibility [100]. This alternative to classical exact visibility computation accelerate rendering times when precise shadow area determination is not needed. Instead of the previously defined scenes, we used a model of several trees (Figure A.5) representative of some applications where an average representation of light transfers may be sufficient. It is also a good test case for the volumetric visibility algorithm : the set of leaves being a good approximation of a turbid media. We calculated the images using each clustering algorithm and then compared the difference with a reference image obtained using exact surface visibility.

A.5 Results

In this section, we present a number of observations drawn from the tests explained previously. We begin with an analysis of the quality aspects of the simulations, looking at the evolution of our error measure with the user-defined error tolerance. We then consider in more detail the capacity of different

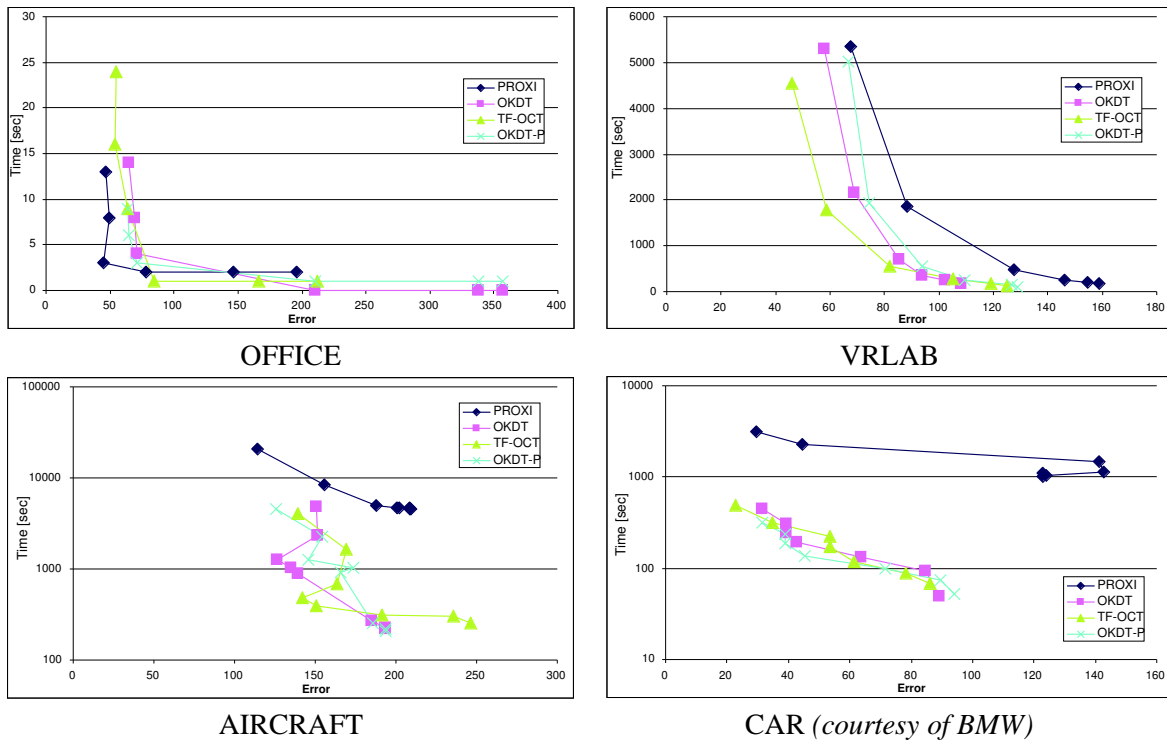


FIG. A.4: Time-error curves

clustering techniques to assist visibility calculations, with a particular emphasis on computational efficiency.

We ran all algorithms on a Silicon Graphics computer (MIPS R10000 at 250 MHZ) with 4GB of memory.

A.5.1 Evolution of Solution Error

Recall from Section A.3 that a major demand on the clustering mechanism (together with the chosen refinement strategy) is that the evolution of computation time and solution quality should be regular and monotonic as the user changes the error tolerance (Figure A.2). Ideally, this would result in a very regular and monotonic time/error curve. Such curves are presented in Figure A.4 for the four test scenes.

Looking at the four Time-error curves, we see two very different types of behaviors : for the OFFICE and VRLAB scenes, all curves are regular and fairly monotonic, whereas for AIRCRAFT and CAR the variation of error is more erratic. Indeed, looking at a plot of error as a function of the user-supplied error tolerance, we find that the error in the solution does not always decrease as we reduce the tolerance. Because of limited space, we present only the error plots for AIRCRAFT and CAR (Figure A.2) ; the corresponding curves for OFFICE and VRLAB are monotonically decreasing.

Why can the error increase when we decrease our tolerance ? this unfriendly behavior occurs when links refined as a result of the error tolerance change produce a less accurate representation of radiosity exchanges. This is largely a question of refinement criteria, but is also influenced by the clustering strategy, as well as the distribution of objects in the scene. We observe that our scenes can be classified into two types : AIRCRAFT and CAR consist of many small polygons, because of a previous tessellation of the objects. VRLAB and OFFICE, on the other hand, contain objects of varying size, from large walls to small furniture components. Based on our experience and the

results of the above experiments, we observe that clustering algorithms have more difficulty with the first type of scene (“polygon soup”). This is especially true of AIRCRAFT because 3D space is very densely populated, resulting in many interactions between clusters which are not separated by a significant distance. These interactions also present a particular challenge to the refinement criterion.

As an illustration of the typical errors created in an approximate solution for such scenes, consider the images in Figure A.3. The two approximate solutions have a similar error under our measure, yet they appear quite different visually. The solution using TF-OCT clusters exhibits marked radiosity variations along axis-aligned boundaries, corresponding to the octree cells; on the other hand, the solution using PROXI shows a high variance of radiosity and a speckle pattern, due to the fact that nearby small objects can belong to many different and overlapping clusters, with markedly different radiosities.

A.5.2 Performance and Visibility Calculation with Clusters

We now consider the performance behavior of the clustering strategies in terms of construction time and as auxiliary structures for visibility calculations.

Construction Time

As explained in Section A.2.2, bottom-up construction is a very expensive process since it amounts to an optimization procedure.

Observed computation times for the construction of the cluster hierarchies support this prediction: the PROXI clustering strategy construction is always much slower than OKDT and TF-OCT, which operate top-down on simple recursive subdivision schemes. These two techniques always take less than 1% of the PROXI time. Interestingly, OKDT-P takes between 20% and 80% of the PROXI time, depending on the distribution of objects in the scene.

Approximate Visibility Calculations

The acceleration of visibility calculations using *equivalent extinction properties* of the clusters has been proposed by Sillion[100]. In this approach, the transmittance factor between two points in the scene is evaluated by considering the entire segment between the points, and its intersections with all clusters, then combining the corresponding attenuation values, in an analogy with partially absorbing volumes. This method, also adopted by Christensen *et al.* [21], is often faster than true ray casting using the surfaces, because of the smaller number of clusters and the ease of computation of ray-cluster intersections.

We computed approximate visibility using clusters in a scene dominated by direct lighting (from the sun), as shown in Figure A.5. In this case, the shadow pattern on the floor is essentially an “X-ray image” of the cluster hierarchy, which greatly helps in the comprehension of the cluster distribution.

We observe a clear hierarchy in terms of shadow quality, in the order PROXI (best, notice high quality of trunk shadows), OKDT-P, OKDT, TF-OCT (poorest). Please see images in color section. This is consistent with the intuitive notion that PROXI starts from the objects and build clusters bottom-up, thereby building clusters that are very tight around the objects.

OKDT (and TF-OCT even more so) exhibits some incorrect shadows of large, blocky clusters, due to the constraints in the spatial subdivision. In this respect, OKDT-P effectively improves on OKDT, with a better fit around the objects and more precise shadows.

Ray Casting Acceleration

Interestingly, the computation times shown in Figure A.5 increase with the quality of the shadows. This is consistent with the general observation that hierarchical structures with lower branching factors have more hierarchical levels and perform better for ray tracing acceleration.

This reasoning is supported by the analysis of cluster statistics on our test scenes. Figure A.6 shows the variation of the following three quantities with the clustering technique, for each test scene :

- total number of clusters
- average number of child elements per cluster
- performance of ray tracing acceleration. This is measured by shooting a large number (100,000) of random rays through the scene and computing ray-surface intersections.

We first observe an obvious inverse correlation between the total number of clusters and the average number of children. In addition, TF-OCT has the largest branching factor for the cluster hierarchy because of its octal subdivision scheme. OKDT also has a fairly high number of children on average, because its construction mechanism offers no way to control this branching factor. Conversely, PROXI has a built-in mechanism limiting the number of children of any given cluster (this operates by grouping objects into overlapping sub-clusters). Therefore it exhibits the lowest branching factor. OKDT-P is intermediate, as expected, because by construction it avoids clusters with many children, handing them to the PROXI clusterizer. Still it avoids the overall large number of clusters of PROXI. A consistent best performer in terms of acceleration is therefore OKDT-P.

Finally, we note the conflicting nature of the two desires for (a) efficient ray tracing acceleration and (b) suitability for radiosity calculations (compare Figure A.6 and Figure A.4).

A.6 Conclusions and Future Work

We have presented an experimental analysis of clustering algorithms for hierarchical radiosity. A taxonomy of clustering algorithms was proposed, followed by a set of requirements for a good clustering algorithm. Guided by these requirements, we developed an experimental methodology based on an image-space quality measure. Extensive tests were run on scenes for the most part developed in real-world application contexts.

Drawing concrete conclusions from experimental tests such as those performed here is always a delicate task. Nonetheless, there are certain elements which we believe are clear enough to be singled out :

Clustering works well in many cases : in particular, for scenes containing objects of different sizes and a sufficient number of large initial surfaces (walls, floors etc.), all the clustering algorithms tested appear to perform well. The Time-error graphs are smooth and monotonic for these cases, presenting the user with an intuitive time-quality tradeoff.

For scenes containing many small objects (“polygon soup”), existing clustering algorithms are less well-behaved. In particular, more time spent computing a solution does not always result in higher quality (see Section A.5.1). This is even more troublesome since the scenes in question are typical of industrial “real-world” models, which are often the result of a fine tessellation of some unspecified and unrecoverable modeling format. It is clear that a new approach is required to treat such models, in order to build a hierarchy that follows the definition of objects. Reconstruction of individual objects is possible based on connectivity and surface properties, and multi-resolution object models could be developed to provide a hierarchy of representations.

Of the clustering algorithms tested, the hierarchical bounding volumes (PROXI) approach seems to have the most predictable behavior in almost all cases. In particular, the Time-error curve is almost always monotonic and smooth. In addition, due to the nature of construction, it fits objects more

tightly, which is a desirable property for clustering. However the overhead for PROXI is significant if not prohibitive in most cases : a much longer construction time (compared to all others tested), longer solution times, and in some cases a higher absolute error for very approximate simulations.

In terms of ray-casting cost, it appears that OKDT-P is the most rapid structure. Thus, if ray-casting cost is an issue (for example in interactive updates where efficiency is paramount), this may be the clustering algorithm of choice.

Finally, in terms of the quality of approximate visibility, a clear hierarchy was found with the following order PROXI (best), OKDT-P, OKDT, TF-OCT (poorest).

We hope these first conclusions will be useful to researchers and developers who wish to use clustering for hierarchical radiosity. Clearly, much remains to be done in this domain.

The error metric adopted for our tests is one of many possibilities. It is evident that different applications have different notions of error (for example in lighting design where an exact measure of energy prevails over image quality), and these different requirements will lead to different choices for clustering. These issues must be further investigated.

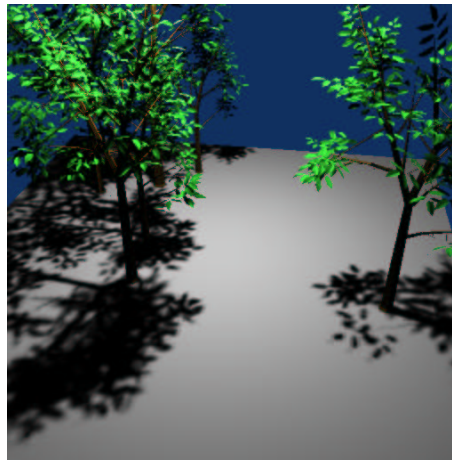
The initial, first-order, classification of scene “type” with respect to their behavior in the context of a clustering algorithm is an interesting avenue of research. Ideally, extensive experimentation would allow us to determine which algorithm is suitable for a given scene. This is however a very ambitious task, so even initial results would be worthy of further research.

The development of a novel clustering approach treating scenes containing many small unrelated polygons is also an interesting challenge.

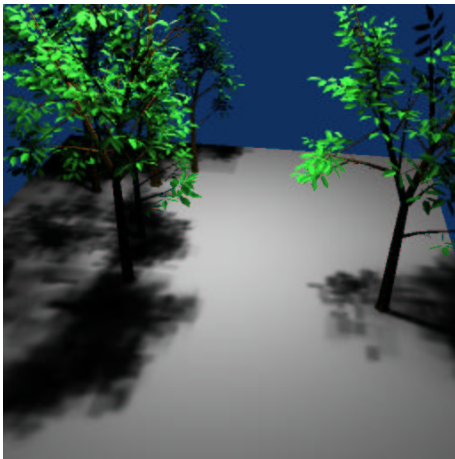
To conclude, we believe that our analysis has shown the utility of clustering for many cases, identified some weaknesses of current algorithms and identified certain important properties of each algorithm with respect to their suitability for different tasks.

A.7 Acknowledgments

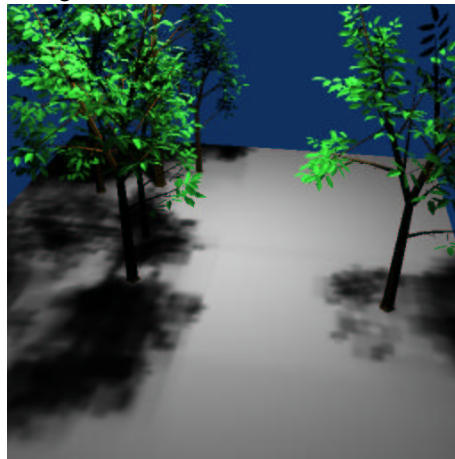
This work was supported by the European Union under the Esprit Long Term Research contract # 24944 “*ARCADE : making radiosity usable*” [5]. Scenes were kindly provided by Lightwork Design Ltd, Fraunhofer Institute for Computer Graphics and BMW. iMAGIS is a joint research project of CNRS/INRIA/UJF/INPG.



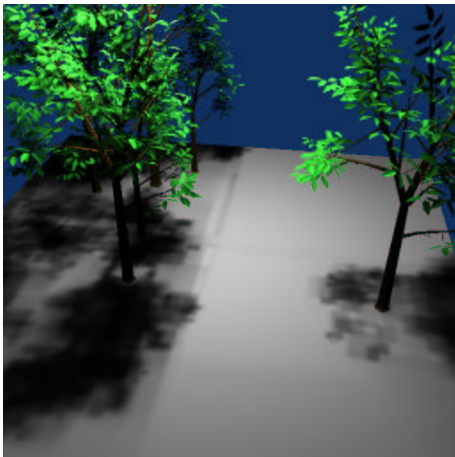
Reference image.



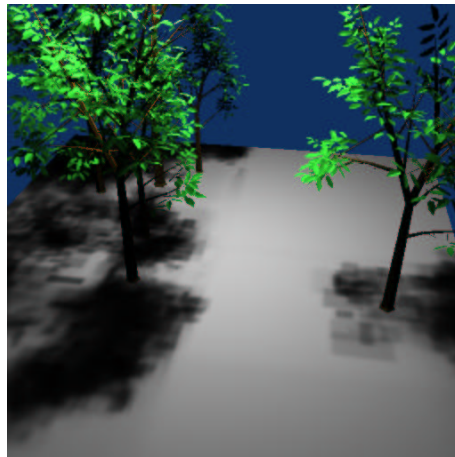
PROXI (629 s)



OKDT-P (402 s)

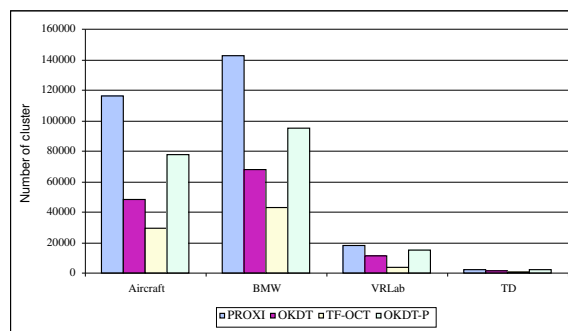


OKDT (281 s)

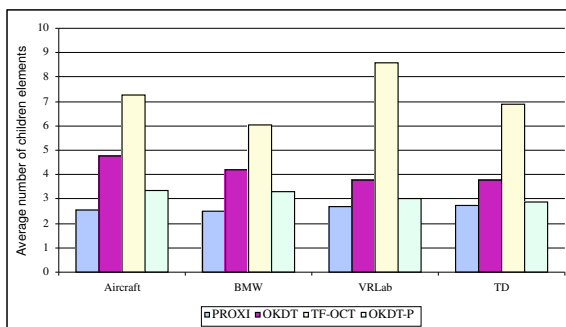


TF-OCT (166 s)

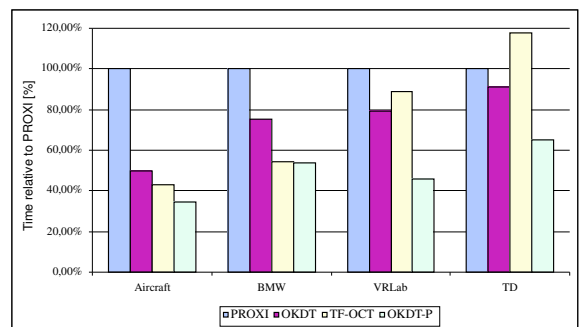
FIG. A.5: Influence of the clustering method on approximate visibility calculations.



Number of clusters



Avg number of children



Ray tracing time

FIG. A.6: Statistics on the cluster hierarchies

TABLE DES MATIÈRES

Introduction	5
I Travaux antérieurs et motivation	9
1 Illumination globale par la méthode de radiosité	11
1.1 Introduction à la simulation de l'éclairage	11
1.1.1 Définitions et notations	12
1.1.2 L'équation du rendu	15
1.2 La méthode de radiosité	18
1.2.1 Discrétisation	19
1.2.2 Radiosité progressive	21
1.2.3 Radiosité hiérarchique	22
1.2.4 Regroupement des surfaces	25
1.2.5 Utilisation d'autres bases de fonctions	28
1.2.6 Affichage de la solution	30
2 Illumination pour les animations	33
2.1 Propriétés de l'irradiance vue comme une fonction du temps	34
2.1.1 Une brève étude des variations d'éclairage au cours d'une animation	34
2.1.2 Continuité temporelle de l'éclairage	36
2.2 Éclairage des animations	37
2.2.1 Méthodes dynamiques	38
2.2.2 Méthodes statiques	41
2.3 Conclusion	43
II Radiosité hiérarchique spatiale et temporelle	45
3 Radiosité spatiale et temporelle	47

3.1	Illumination diffuse 4D : formulation continue	48
3.1.1	Hypothèses et notations	48
3.1.2	Mise en équation	49
3.2	Discrétisation	50
3.3	Un exemple de discrétisation : Base d'éléments constants	51
3.3.1	Équations discrètes de la radiosit� 4D	51
3.3.2	Propri�t�s du facteur de forme 4D	52
3.4	Discussion	53
4	Algorithme de r�solution hi�rarchique	55
4.1	Hi�rarchie des �l�ments	56
4.1.1	Choix d'une structure	56
4.1.2	M�canisme de subdivision	56
4.1.3	Maintenance de la coh�rence hi�rarchique	57
4.1.4	Affichage de la solution	58
4.2	�changes �nerg�tiques	59
4.2.1	Structure de la hi�rarchie des liens	59
4.2.2	Raffinement des liens	59
4.2.3	D�termination de la visibilit�	61
4.3	Discussion	62
5	R�sultats et interpr�tation	65
5.1	Efficacit� de la m�thode	65
5.1.1	Exp�rimentations	65
5.1.2	R�sultats	66
5.1.3	Commentaires	66
5.2	Mise en �vidence des points n�cessitant am�lioration	68
5.2.1	Calcul sur des sc�nes complexes	68
5.2.2	Discontinuit�s dans l'�clairage indirect	68
5.3	Conclusion	69
III	Mise en pratique de l'algorithme de radiosit� hi�rarchique spatiale et temporelle	71
6	Regroupement des surfaces	73
6.1	Construction paresseuse de la hi�rarchie de groupes	74
6.1.1	Principe de la construction	74
6.1.2	Crit�res pour le choix d'un algorithme de construction	74
6.1.3	Subdivision spatiale d'un groupe	76
6.1.4	Subdivision temporelle d'un groupe	76
6.2	Utilisation des groupes	77
6.2.1	�mission depuis un groupe	78
6.2.2	R�ception de l'�nergie dans un groupe	79
6.2.3	�chantillonnage des groupes	80
6.3	R�sultats et discussion	81

7	Utilisation de multi-ondelettes	83
7.1	Ondelettes et multi-ondelettes	84
7.1.1	Analyse multirésolution	84
7.1.2	Les multi-ondelettes : une base orthogonale de $\mathcal{L}_2([0, 1])$	86
7.2	Application à la radiosité spatiale et temporelle	88
7.2.1	Implémentation	88
7.2.2	Interactions entre éléments	89
7.2.3	Un nouvel oracle de raffinement	92
7.2.4	Mise à jour de la hiérarchie	93
7.3	Résultats expérimentaux	96
7.3.1	Amélioration de la continuité temporelle	96
7.3.2	Performances de l'algorithme	97
7.3.3	Conclusion	98
8	Ordonnancement des calculs	101
8.1	Principe de l'ordonnancement des calculs	102
8.2	Tâches à effectuer et contraintes temporelles	102
8.2.1	Recensement des liens	102
8.2.2	Raffinement des interactions et collecte des irradiances	103
8.2.3	Mise à jour : Descente (<i>Push</i>)	104
8.2.4	Mise à jour : Remontée (<i>Pull</i>)	105
8.2.5	Bilan	105
8.3	Mécanisme d'ordonnancement des calculs	106
8.4	Gestion des éléments et mécanisme de chargement	106
8.4.1	Chargement et sauvegarde des éléments	107
8.4.2	Création des éléments	108
8.4.3	Traduction des identificateurs en pointeurs	109
8.5	Conclusion	109
8.5.1	Résultats	109
8.5.2	Discussion	109
	Conclusion	113
A	Etude des algorithmes de regroupement	117
A.1	Introduction and Motivation	119
A.2	A Taxonomy of Clustering Algorithms	119
A.2.1	Data Structure Choices	120
A.2.2	Construction Algorithms	121
A.2.3	Improved Approaches	122
A.3	Requirements for a Clustering Algorithm	122
A.3.1	Requirements Regarding the Quality of the Results	123
A.3.2	Efficiency Considerations	123
A.4	Experimental methodology for evaluation clustering algorithms	124
A.4.1	Methodology	124
A.4.2	Clustering Strategies Studied	125
A.4.3	Test Scenes Chosen	126
A.4.4	Tests Performed	126
A.5	Results	127

A.5.1	Evolution of Solution Error	128
A.5.2	Performance and Visibility Calculation with Clusters	129
A.6	Conclusions and Future Work	130
A.7	Acknowledgments	131
Table des matières		135
Table des figures		139
Table des algorithmes		141
Liste des tableaux		142
Bibliographie		143

TABLE DES FIGURES

1.1	Définition de l'angle solide	12
1.2	Direction en coordonnées polaires	13
1.3	Notations pour l'équation 1.7	16
1.4	Distinction entre B , B^\perp , \tilde{B}	20
1.5	Diversité de l'importance des échanges ¹	22
1.6	Hierarchie utilisant un arbre quaternaire	23
1.7	Mise à jour hiérarchique	24
1.8	Exemple d'occlusion triviale	25
1.9	Exemples de hiérarchies de groupes	26
1.10	Anisotropie des groupes	27
1.11	Comparaison entre approximation uniforme et approximation linéaire	28
1.12	Construction d'une base de fonctions sur \mathbb{R}^2	29
1.13	Polynômes orthogonaux de Legendre	29
1.14	Exemples de résultats de <i>récoltes finales</i>	31
2.1	Deux scènes de test	35
2.2	Énergie lumineuse émise en fonction de la durée d'un lien	35
2.3	Discontinuité temporelle due à un effet de seuil	36
2.4	Ré-éclairage interactif d'une scène	38
2.5	Mise à jour de l'affichage du "cache de rendu"	39
2.6	Correction de l'énergie due à un objet en mouvement	40
2.7	Mise à jour dynamique d'un lien	41
2.8	Faisceaux joignant deux boîtes englobantes	42
3.1	Notations pour la description des animations	49
4.1	Exemple de hiérarchie spatiale et temporelle	57
4.2	Visualisation du maillage spatial et temporel à deux dates différentes	58
4.3	Exemple de configuration permettant une optimisation	60
4.4	Illustration de l'algorithme 4.3	63

5.1	Quelques images représentatives des scènes sur lesquelles nous avons effectué nos tests.	67
5.2	Exemple de discontinuités spatiales.	69
5.3	Exemple de discontinuité temporelle	70
6.1	Comparaison KD-arbre/boîtes englobantes	75
6.2	Différentes possibilités de subdivision d'un groupe	77
6.3	Échantillonnage spatial d'un groupe	80
6.4	Scènes de tests pour le regroupement	82
7.1	Fonctions d'échelle pour la base de Haar de résolution 2	85
7.2	Ondelettes de Haar de résolution 1	85
7.3	Base de Daubechies D_4	86
7.4	Multi-ondelettes \mathcal{M}_2	87
7.5	Fonctions d'échelle de la base \mathcal{M}_4	88
7.6	Variations temporelles de la radiosité d'un élément	96
7.7	Comparaison des discontinuités temporelles à $t = \frac{1}{2}$.	97
7.8	Quelques images de notre animation testant la continuité de l'éclairage	99
7.9	Quelques images de notre animation testant l'utilisation combinée des groupes et des ondelettes	100
8.1	Repérage des intervalles de temps correspondant à des éléments obsolètes	108
8.2	Évolution de la consommation mémoire vive en fonction de la résolution des agendas	110
A.1	The four test scenes.	125
A.2	Error curves as function of user-supplied tolerance	127
A.3	Example solutions exhibiting different (visual) forms of error.	127
A.4	Time-error curves	128
A.5	Influence of the clustering method on approximate visibility calculations.	132
A.6	Statistics on the cluster hierarchies	133

TABLE DES ALGORITHMES

4.1	Mise à jour récursive de la hiérarchie	58
4.2	Évaluation de la visibilité dans une hiérarchie de boîtes englobantes en mouvement .	61
4.3	Construction d'une hiérarchie de boîtes englobantes (d'après [21])	62
6.1	Mécanisme de subdivision spatiale d'un groupe (d'après [21])	76
6.2	Mécanisme de subdivision temporelle d'un groupe	77
7.1	Mise à jour récursive de la hiérarchie	95
8.1	Tâche atomique de recensement des liens	103
8.2	Tâche atomique de raffinement et de collecte d'un lien	103
8.3	Tâche atomique de mise à jour descendante	104
8.4	Tâche atomique de mise à jour ascendante	105
8.5	Algorithme hiérarchique réordonné	107

LISTE DES TABLEAUX

1.1	Récapitulatif des grandeurs et unités	15
2.1	Proportion d'énergie constante durant toute l'animation	35
5.1	Performances de l'algorithme de radiosité 4D hiérarchique	66
6.1	Résultats comparatifs pour l'utilisation du regroupement	81
7.1	Comparatifs des performances (calcul image par image, base de Haar, \mathcal{M}_2), scène simple	97
7.2	Résultats expérimentaux, scène complexe	98
8.1	Contraintes sur l'exécution des différentes tâches	105
8.2	Consommation mémoire détaillée, pour différentes scènes de test	110

BIBLIOGRAPHIE

- [1] ADELSON (E. H.) et BERGEN (J. R.), « The Plenoptic Function and the Elements of Early Vision », *Computational Models of Visual Processing*, 1991. (cité à la page 14)
- [2] AIREY (John M.), ROHLF (John H.) et FREDERICK P. BROOKS (Jr.), « Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments », dans *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, volume 24, p. 41–50, mars 1990. (cité à la page 37)
- [3] ALPERT (B.), BEYLKIN (G.), COIFMAN (R.) et ROKHLIN (V.), « Wavelet-like bases for the fast solution of second-kind integral equations », *SIAM Journal on Scientific Computing*, 14, n° 1, 1993, p. 159–184. (cité à la page 84)
- [4] ALPERT (Bradley K.), « A Class of Bases in L^2 for the Sparse Representation of Integral Operators », *SIAM Journal on Mathematical Analysis*, 24, n° 1, janvier 1993, p. 246–262. (cité aux pages 28, 84, 86)
- [5] *ARCADE : making radiosity usable*, European Union. Esprit Long Term Research project #24944. <http://www-imagis.imag.fr/ARCADE>. (cité à la page 131)
- [6] ARVO (James), Linear Operators and Integral Equations in Global Illumination, dans *ACM SIGGRAPH '93 Course Notes - Global Illumination*, Chapitre 2, p. 1–21, 1993. (cité à la page 18)
- [7] ———, « The Role of Functional Analysis in Global Illumination », dans HANRAHAN (P. M.) et PURGATHOFER (W.), éditeurs, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, p. 115–126, New York, NY, 1995, Springer-Verlag. (cité aux pages 12, 15)
- [8] BAUM (Daniel R.), MANN (Stephen), SMITH (Kevin P.) et WINGET (James M.), « Making Radiosity Usable : Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions », dans *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, p. 51–60, juillet 1991. (cité à la page 30)
- [9] BAUM (Daniel R.), RUSHMEIER (Holly E.) et WINGET (James M.), « Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors », dans *Computer Graphics (ACM SIGGRAPH '89 Proceedings)*, volume 23, p. 325–334, juillet 1989. (cité à la page 21)

- [10] BAUM (Daniel R.), WALLACE (John R.), COHEN (Michael F.) et GREENBERG (Donald P.), « The Back-Buffer Algorithm : An Extension of the Radiosity Method to Dynamic Environments », *The Visual Computer*, 2, n° 5, septembre 1986, p. 298–306. (cité à la page 42)
- [11] BEKAERT (Philippe) et WILLEMS (Yves D.), « Error Control for Radiosity », dans *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, p. 153–164, New York, NY, 1996, Springer-Verlag/Wien. (cité aux pages 30, 92)
- [12] ———, « Hirad : A Hierarchical Higher Order Radiosity Implementation », dans *Proceedings of the Twelfth Spring Conference on Computer Graphics (SCCG '96)*, Bratislava, Slovakia, juin 1996, Comenius University Press. (cité aux pages 30, 92)
- [13] BERGMAN (L. D.), FUCHS (H.), GRANT (E.) et SPACH (S.), « Image Rendering by Adaptive Refinement », dans EVANS (David C.) et ATHAY (Russell J.), éditeurs, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, p. 29–37, août 1986. (cité à la page 38)
- [14] BESUIEVSKY (Gonzalo) et PUEYO (Xavier), « Animating Radiosity Environments Through the Multi-Frame Lighting Method », *Journal of Visualization and Computer Animation*, 12, 2001, p. 93–106. (cité à la page 41)
- [15] BESUIEVSKY (Gonzalo) et SBERT (Mateu), « The Multi-Frame Lighting Method : A Monte Carlo Based Solution for Radiosity in Dynamic Environments », dans *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, p. 185–194, New York, NY, 1996, Springer-Verlag/Wien. (cité aux pages 36, 41)
- [16] BISHOP (Gary), FUCHS (Henry), MCMILLAN (Leonard) et SCHER ZAGIER (Ellen J.), « Frameless Rendering : Double Buffering Considered Harmful », *Computer Graphics*, 28, n° Annual Conference Series, 1994, p. 175–176. (cité à la page 38)
- [17] CAZALS (Frédéric), DRETTAKIS (George) et PUECH (Claude), « Filtering, Clustering and Hierarchy Construction : a New Solution for Ray Tracing Very Complex Environments », dans POST (F.) et GÖBEL (M.), éditeurs, *Computer Graphics Forum (Proc. of Eurographics '95)*, volume 15, Maastricht, the Netherlands, septembre 1995. (cité à la page 120)
- [18] CAZALS (Frédéric) et PUECH (Claude), « Bucket-like Space Partitioning Data Structures with Applications to Ray-Tracing », dans *Proceedings of the 13th International Annual Symposium on Computational Geometry (SCG-97)*, p. 11–20, New York, juin 4–6 1997, ACM Press. (cité aux pages 120, 121)
- [19] CHEN (Shenchang Eric), « Incremental Radiosity : An Extension of Progressive Radiosity to an Interactive Image Synthesis System », dans *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, p. 135–144, août 1990. (cité aux pages 39, 40)
- [20] CHEN (Shenchang Eric) et WILLIAMS (Lance), « View Interpolation for Image Synthesis », *Proceedings of SIGGRAPH 93, Anaheim, California.*, août 1993, p. 279–288, ISBN 0-201-58889-7. (cité à la page 42)
- [21] CHRISTENSEN (Per H.), LISCHINSKI (Dani), STOLLNITZ (Eric J.) et SALESIN (David H.), « Clustering for Glossy Global Illumination », *ACM Transactions on Graphics*, 16, n° 1, janvier 1997, p. 3–33. (cité aux pages 26, 62, 75, 76, 121, 123, 129, 141)
- [22] COHEN (Michael), CHEN (Shenchang Eric), WALLACE (John R.) et GREENBERG (Donald P.), « A Progressive Refinement Approach to Fast Radiosity Image Generation », dans *Computer Graphics (ACM SIGGRAPH '88 Proceedings)*, volume 22, p. 75–84, août 1988. (cité aux pages 20, 21)
- [23] COHEN (Michael F.) et GREENBERG (Donald P.), « The Hemi-Cube : A Radiosity Solution for Complex Environments », dans BARSKY (B. A.), éditeur, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, p. 31–40, 1985. (cité à la page 21)

- [24] CUNY (François), *Radiosité à base d'ondelettes sur des surfaces paramétriques*, Thèse de doctorat, Ecole Doctorale IAE+M, Institut National Polytechnique de Lorraine, octobre 2000. (cité à la page 19)
- [25] CUNY (François), ALONSO (Laurent) et HOLZSCHUCH (Nicolas), « A Novel Approach Makes Higher Order Wavelets Really Efficient for Radiosity », dans *Computer Graphics Forum (Proc. Eurographics 2000)*, volume 19, p. C-99–C-108, 2000, Disponible à <http://www.loria.fr/holzschu/Publications/paper20.pdf>. (cité aux pages 30, 92)
- [26] DAMEZ (Cyrille) et SILLION (François X.), « Space-Time Hierarchical Radiosity », dans *Rendering Techniques '99*, p. 235–246, New York, NY, 1999, Springer Wien. (cité à la page 7)
- [27] DAMEZ (Cyrille), SILLION (François X.) et HOLZSCHUCH (Nicolas), « Space-Time Hierarchical Radiosity with Clustering and Higher-Order Wavelets », dans *Short Papers Proceedings of Eurographics 2001*, septembre 2001. (cité à la page 7)
- [28] DAUBECHIES (Ingrid), « Orthogonal bases of compactly supported wavelets », *Communication on Pure and Applied Mathematics*, 41, 1988, p. 909–996. (cité à la page 86)
- [29] DORSEY (Julie O'B.), *Computer Graphics Techniques for Opera Lighting Design and Simulation*, Ph.D. thesis, Cornell University, Ithaca, NY, janvier 1993. (cité à la page 37)
- [30] DRETTAKIS (George) et SILLION (François X.), « Interactive Update of Global Illumination Using a Line-Space Hierarchy », dans *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, volume 31, p. 57–64, 1997. (cité aux pages 41, 123)
- [31] FAIRCHILD (Mark D.), *Color Appearance Models*, Addison Wesley, Chapitre 3, p. 90–93, 1998, p. 90–93. (cité à la page 126)
- [32] FINKELSTEIN (Adam), JACOBS (Charles E.) et SALESIN (David H.), « Multiresolution Video », dans *Computer Graphics, Annual Conference Series : (ACM SIGGRAPH '96 Proceedings)*, p. 281–290, 1996. (cité à la page 56)
- [33] FOLEY (James D.), van DAM (Andries), FEINER (Steven K.) et HUGHES (John F.), *Computer Graphics, Principles and Practice, Second Edition*, Reading, Massachusetts, Addison-Wesley, 1990. (cité à la page 11)
- [34] FORSYTH (David A.), YANG (Chien) et TEO (Kim), « Efficient Radiosity in Dynamic Environments », dans *Fifth Eurographics Workshop on Rendering*, p. 313–323, Darmstadt, Germany, juin 1994. (cité aux pages 40, 41)
- [35] GABOR (D.), « Theory of Communication », *Journal of the Institute for Electrical Engineers*, 93, 1946, p. 429–439. (cité à la page 28)
- [36] GAEDE (Volker) et GÜNTHER (Oliver), « Multidimensional access methods », *ACM Computing Surveys*, 30, n° 2, 1998, p. 170–231. (cité à la page 61)
- [37] GEORGE (David W.), SILLION (François X.) et GREENBERG (Donald P.), « Radiosity Redistribution for Dynamic Environments », *IEEE Computer Graphics and Applications*, 10, n° 4, juillet 1990, p. 26–34. (cité aux pages 34, 39, 40)
- [38] GIBSON (Simon) et HUBBOLD (R. J.), « Efficient Hierarchical Refinement and Clustering for Radiosity in Complex Environments », *Computer Graphics Forum*, 15, n° 5, décembre 1996, p. 297–310. (cité aux pages 30, 120, 121, 123)
- [39] GLASSNER (Andrew S.), « Space Subdivision For Fast Ray Tracing », *IEEE Computer Graphics & Applications*, 4, n° 10, octobre 1984, p. 15–22. (cité à la page 26)
- [40] _____, « Spacetime Ray Tracing for Animation », *IEEE Computer Graphics and Applications*, 8, n° 2, mars 1988, p. 60–70. (cité à la page 41)

- [41] _____, « A Model for Fluorescence and Phosphorescence », dans SAKAS (Georgios) et MÜLLER (Stephan), éditeurs, *Photorealistic Rendering Techniques, proceedings of the fifth Eurographics Workshop on Rendering*, p. 60–70, Springer-Verlag, 1994. (cité à la page 17)
- [42] GOLDSMITH (Jeffrey) et SALMON (John), « Automatic Creation of Object Hierarchies for Ray Tracing », *IEEE Computer Graphics & Applications*, 7, n° 5, mai 1987, p. 14–20. (cité aux pages 26, 75, 124)
- [43] GOLUB (Gene H.) et LOAN (Charles F. Van), *Matrix computations*, Baltimore, MD, USA, The Johns Hopkins University Press, Johns Hopkins Studies in the Mathematical Sciences, third édition, 1996. (cité à la page 21)
- [44] GORAL (Cindy M.), TORRANCE (Kenneth E.), GREENBERG (Donald P.) et BATTAILE (Bennett), « Modelling the Interaction of Light Between Diffuse Surfaces », dans *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, p. 212–222, juillet 1984. (cité aux pages 17, 18, 20, 21)
- [45] GORTLER (Steven J.), GRZESZCZUK (Radek), SZELISKI (Richard) et COHEN (Michael F.), « The Lumigraph », dans *Computer Graphics Proceedings, Annual Conference Series, 1996 (ACM SIGGRAPH '96 Proceedings)*, p. 43–54, 1996. (cité à la page 14)
- [46] GORTLER (Steven J.), SCHRODER (Peter), COHEN (Michael F.) et HANRAHAN (Pat), « Wavelet Radiosity », dans *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, p. 221–230, 1993. (cité aux pages 20, 28, 84, 88)
- [47] GOURAUD (Henri), « Continuous Shading of Curved Surfaces », *IEEE Transactions on Computers*, C-20, n° 6, 1971, p. 623–629. (cité à la page 30)
- [48] GRANIER (Xavier), *Contrôle automatique de qualité pour l'éclairage global*, Ph.D. thesis, Université Joseph Fourier, novembre 2001. (cité à la page 39)
- [49] GRANIER (Xavier) et DRETTAKIS (George), « Controlling Memory Consumption of Hierarchical Radiosity with Clustering », dans *Proceedings of Graphics Interface '99*, p. 58–65, San Francisco, CA, juin 1999, Morgan Kaufman. (cité à la page 30)
- [50] _____, « Incremental Updates for Rapid Glossy Global Illumination », dans *Eurographics 2001*, 2001. (cité à la page 41)
- [51] GRANIER (Xavier), DRETTAKIS (George) et WALTER (Bruce), « Fast Global Illumination Including Specular Effects », dans PEROCHE (B.) et RUSHMEIER (H.), éditeurs, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, p. 47–58, New York, NY, 2000, Springer Wien. (cité à la page 115)
- [52] HAAR (Alfred), « Zur Theorie der orthogonalen Funktionensysteme », *Mathematische Annalen*, 69, 1910, p. 331–371. (cité à la page 28)
- [53] HAINES (Eric) et WALLACE (John), « Shaft culling for efficient ray-traced radiosity », dans *Eurographics Workshop on Rendering*, p. 122–138, 1991. (cité aux pages 41, 60)
- [54] HALTON (J.) et WELLER (G.), « Algorithm 247 : Radicalinverse quasi-random point sequence », *Communication of the ACM*, 7, n° 12, 1964, p. 701–702. (cité à la page 80)
- [55] HANRAHAN (Pat), SALZMAN (David) et AUPPERLE (Larry), « A Rapid Hierarchical Radiosity Algorithm », dans *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, p. 197–206, juillet 1991. (cité aux pages 22, 40, 53, 55, 56, 62, 68, 73, 84, 124, 126)
- [56] HASENFRATZ (Jean-Marc), DAMEZ (Cyrille), SILLION (François X.) et DRETTAKIS (Georges), « A Practical Analysis of Clustering Strategies for Hierarchical Radiosity », dans *Computer Graphics Forum (Proc. Eurographics '99)*, volume 18, p. C–221–C–232, septembre 1999. (cité aux pages 26, 62, 74, 75)

- [57] HASENFRATZ (Jean-Marc), DAMEZ (Cyrille), SILLION (François) et DRETTAKIS (George), « A Practical Analysis of Clustering Strategies for Hierarchical Radiosity », dans *Computer Graphics Forum (Proc. Eurographics '99)*, volume 18, septembre 1999. (cité à la page 117)
- [58] HECKBERT (Paul), *Simulating Global Illumination Using Adaptive Meshing*, Ph.D. thesis, Technical Report, Univ. of California, Berkeley, juin 1991. (cité à la page 14)
- [59] _____, « Discontinuity Meshing for Radiosity », dans *Third Eurographics Workshop on Rendering*, p. 203–226, Bristol, UK, mai 1992. (cité à la page 20)
- [60] _____, « Radiosity in Flatland », dans *Computer Graphics Forum (Eurographics '92)*, volume 11, p. C181–C192, Cambridge, UK, septembre 1992. (cité à la page 42)
- [61] _____, Finite Element Methods for Radiosity, dans *ACM SIGGRAPH '93 Course Notes - Global Illumination*, Chapitre 5, p. 1–7, 1993. (cité à la page 18)
- [62] _____, Introduction to Finite Element Methods, dans *ACM SIGGRAPH '93 Course Notes - Global Illumination*, Chapitre 3, p. 1–10, 1993. (cité à la page 18)
- [63] HECKBERT (Paul) et WINGET (J. M.), « Finite Element Methods for Global Illumination », Technical Report UCB/CSD 91/643, University of California at Berkeley, Berkeley, CA, 1991. (cité à la page 18)
- [64] HOLZSCHUCH (Nicolas), SILLION (François X.) et DRETTAKIS (Georges), « An Efficient Progressive Refinement Strategy for Hierarchical Radiosity », dans *Fifth Eurographics Workshop on Rendering*, p. 343–357, Darmstadt, Germany, juin 1994. (cité aux pages 18, 23, 25)
- [65] HOWELL (John R.), *A Catalog of Radiation Configuration Factors*, New York, NY, McGraw Hill, 1982. (cité à la page 21)
- [66] KAJIYA (James T.), « The Rendering Equation », dans *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, p. 143–150, août 1986. (cité aux pages 11, 16, 37, 38)
- [67] KLIMASZEWSKI (Krzysztof S.) et SEDERBERG (Thomas W.), « Faster Ray Tracing Using Adaptive Grids », *IEEE Computer Graphics and Applications*, 17, n° 1, janvier 1997, p. 42–51. (cité à la page 120)
- [68] LEVOY (Marc) et HANRAHAN (Pat), « Light Field Rendering », dans *Computer Graphics Proceedings, Annual Conference Series, 1996 (ACM SIGGRAPH '96 Proceedings)*, p. 31–42, 1996. (cité à la page 14)
- [69] LISCHINSKI (Dani), SMITS (Brian) et GREENBERG (Donald P.), « Bounds and Error Estimates for Radiosity », dans *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, p. 67–74, 1994. (cité à la page 22)
- [70] LISCHINSKI (Dani), TAMPIERI (Filippo) et GREENBERG (Donald P.), « Discontinuity Meshing for Accurate Radiosity », *IEEE Computer Graphics and Applications*, 12, n° 6, novembre 1992, p. 25–39. (cité à la page 20)
- [71] MALLAT (Stéphane), *Multiresolution Representations and Wavelets*, Ph.D. thesis, University of Pennsylvania, août 1988. (cité à la page 28)
- [72] MARTIN (Ignacio), PUEYO (Xavier) et TOST (Dani), « Frame-to-Frame Coherent Animation with Two-Pass Radiosity », Rapport Technique IliA 99-08-RR, Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona, Espagne, juin 1999, à paraître dans *IEEE Transactions on Visualization and Computer Graphics*. (cité aux pages 43, 115)
- [73] MARTÍN (Ignacio), *Temporal Coherence in Animation of Global Illumination Environments*, Ph.D. thesis, Universitat Politècnica de Catalunya, 2000. (cité à la page 115)

- [74] MCMILLAN (Leonard) et BISHOP (Gary), « Plenoptic Modeling : An Image-Based Rendering System », *Proceedings of SIGGRAPH 95*, août 1995, p. 39–46, ISBN 0-201-84776-0. Los Angeles, California. (cité à la page 14)
- [75] MEIER (Barbara J.), « Painterly Rendering for Animation », dans *Computer Graphics Proceedings, Annual Conference Series 1996. (ACM SIGGRAPH 1996 Proceedings)*, volume 30, p. 477–484, 1996. (cité à la page 36)
- [76] MERZOUK (Slimane), *Architecture Logicielle et Algorithmes Pour la Resolution de L'Equation de Radiance*, Thèse de doctorat, Institut National Polytechnique de Lorraine, 1997. (cité à la page 19)
- [77] MEYER (Yves), *Ondelettes et Opérateurs, I : Ondelettes, II : Opérateurs de Calderón-Zygmund, III : (with R. Coifman), Opérateurs multilinéaires*, Paris, Hermann, 1990. (cité à la page 28)
- [78] MYSZKOWSKI (Karol), TAWARA (Takehiro), AKAMINE (Hiroyuki) et SEIDEL (Hans-Peter), « Perception-Guided Global Illumination Solution for Animation Rendering », dans *Computer Graphics Proceedings, Annual Conference Series 2001, (ACM SIGGRAPH 2001 Proceedings)*, p. 221–230, 2001. (cité aux pages 36, 42, 115)
- [79] NICODEMUS (F. E.), RICHMOND (J. C.), HSIA (J. J.), GINSBERG (I. W.) et LIMPERIS (T.), « Geometric Considerations and Nomenclature for Reflectance », Monographie 161, National Bureau of Standards (US), octobre 1977. (cité à la page 14)
- [80] NIMEROFF (Jeffrey), DORSEY (Julie) et RUSHMEIER (Holly), « A Framework for Global Illumination in Animated Environments », dans HANRAHAN (P. M.) et PURGATHOFER (W.), éditeurs, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, p. 92–103, New York, NY, 1995, Springer-Verlag. (cité à la page 42)
- [81] ———, « Implementation and Analysis of an Image-Based Global Illumination Framework for Animated Environments », *IEEE Transactions on Visualization and Computer Graphics*, 2, n° 4, décembre 1996, p. 283–298. (cité à la page 42)
- [82] NISHITA (Tomoyuki) et NAKAMAE (Eihachiro), « Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection », dans *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, volume 19, p. 23–30, juillet 1985. (cité à la page 40)
- [83] ORTI (Rachel), *Radiosite Dynamique 2D et Complexe de Visibilité*, Ph.D. thesis, Université Joseph Fourier, juillet 1997. (cité à la page 37)
- [84] ORTI (Rachel), RIVIERE (Stephane), DURAND (Fredo) et PUECH (Claude), « Radiosity for Dynamic Scenes in Flatland with the Visibility Complex », dans *Computer Graphics Forum*, volume 15, p. C237–C248, septembre 1996. (cité à la page 42)
- [85] PATTANAİK (Sumanta N.), FERWERDA (James A.), GREENBERG (Donald A.) et FAIRCHILD (Mark D.), « A Multiscale Model of Adaptation and Spatial Vision for Realistic Imaging », dans *Computer Graphics (ACM SIGGRAPH '98 Proceedings)*, p. 287–298, 1998. (cité à la page 31)
- [86] PERRY (R. L.) et SPECK (E. P.), « Geometric Factors for Thermal Radiation Exchange Between Cows and their Surroundings », *Transactions of the American Society of Agricultural Engineers*, 5, n° 1, 1962, p. 31–37. (cité à la page 21)
- [87] PRAUN (Emil), HOPPE (Hugues), WEBB (Matthew) et FINKELSTEIN (Adam), « Real-Time Hatching », dans *Computer Graphics Proceedings, Annual Conference Series 2001, (ACM SIGGRAPH 2001 Proceedings)*, p. 581–586, 2001. (cité à la page 36)

- [88] PRESS (William), TEUKOLSKY (Saul), VETTERLING (William) et FLANNERY (Brian), *Numerical Recipes in C, second edition*, New York, USA, Cambridge University Press, 1992. (cité à la page 79)
- [89] PUEYO (Xavier), « Diffuse Interreflections. Techniques for Form-Factor Computation : A Survey », *The Visual Computer*, 7, n° 4, juillet 1991, p. 200–209. (cité à la page 21)
- [90] REICHERT (Mark C.), *A Two-Pass Radiosity Method Driven by Lights and Viewer Position*, M.Sc. thesis, Cornell University, Ithaca, NY, janvier 1992. (cité aux pages 30, 43)
- [91] REINHARD (Erik), SMITS (Brian) et HANSEN (Chuck), « Dynamic Acceleration Structures for Interactive Ray Tracing », *Rendering Techniques 2000 : 11th Eurographics Workshop on Rendering*, juin 2000, p. 299–306, ISBN 3-211-83535-0. (cité à la page 61)
- [92] SAMET (Hanan), *Applications of Spatial Data Structures*, Reading, Massachusetts, Addison-Wesley, 1990. (cité à la page 30)
- [93] SCHEEL (Annette), STAMMINGER (M.) et SEIDEL (Hans-Peter), « Thrifty Final Gather for Radiosity », dans *12th EG Workshop on Rendering, "Rendering Techniques 2001"*, p. 1–12, Springer Wien, 2001. (cité à la page 30)
- [94] SCHOEFFEL (F.) et POMI (A.), « Reducing Memory Requirements for Interactive Radiosity Using Movement Prediction », dans *Rendering Techniques '99*, p. 225–234, New York, NY, 1999, Springer Wien. (cité à la page 41)
- [95] SCHRODER (Peter), *Wavelet Algorithms for Illumination Computations*, Ph.D. thesis, Technical Report, Princeton University, Princeton, NJ, novembre 1994. (cité aux pages 73, 85, 88)
- [96] SCHRODER (Peter), GORTLER (Steven J.), COHEN (Michael F.) et HANRAHAN (Pat), « Wavelet Projections for Radiosity », dans *Fourth Eurographics Workshop on Rendering*, Series EG 93 RW, p. 105–114, Paris, France, juin 1993. (cité à la page 20)
- [97] SHAW (Erin), « Hierarchical Radiosity for Dynamic Environments », *Computer Graphics Forum*, 16, n° 2, juin 1997, p. 107–118. (cité à la page 41)
- [98] SIEGEL (Robert) et HOWELL (John R.), *Thermal Radiation Heat Transfer, 3rd Edition*, New York, NY, Hemisphere Publishing Corporation, 1992. (cité à la page 21)
- [99] SILLION (François X.), « Clustering and Volume Scattering for Hierarchical Radiosity Calculations », dans *Fifth Eurographics Workshop on Rendering*, p. 105–117, Darmstadt, Germany, juin 1994. (cité aux pages 25, 27, 41, 73, 74, 119, 120, 123)
- [100] ———, « A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters », *IEEE Transactions on Visualization and Computer Graphics*, 1, n° 3, septembre 1995. (cité aux pages 77, 79, 119, 120, 121, 127, 129)
- [101] SILLION (François X.) et DRETTAKIS (George), « Feature-Based Control of Visibility Error : A Multiresolution Clustering Algorithm for Global Illumination », dans *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, p. 145–152, 1995. (cité aux pages 121, 123)
- [102] SILLION (François X.) et HASENFRATZ (Jean-Marc), « Efficient Parallel Refinement for Hierarchical Radiosity on a DSM Computer », dans *Proceedings of the Third Eurographics Workshop on Parallel Graphics and Visualisation*, Universitat de Girona, Spain, septembre 2000, Disponible à l'adresse <http://www-imagis.imag.fr/Jean-Marc.Hasenfratz/PUBLI/EGWPGV00.html>. (cité à la page 102)
- [103] SILLION (François X.) et PUECH (Claude), « A General Two-Pass Method Integrating Specular and Diffuse Reflection », dans *Computer Graphics (ACM SIGGRAPH '89 Proceedings)*, volume 23, p. 335–344, juillet 1989. (cité aux pages 30, 43)

- [104] _____, *Radiosity and Global Illumination*, San Francisco, CA, Morgan Kaufmann, 1994. (cité aux pages 12, 14, 22, 40)
- [105] SMITS (Brian), ARVO (James) et GREENBERG (Donald), « A Clustering Algorithm for Radiosity in Complex Environments », dans *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, p. 435–442, 1994. (cité aux pages 25, 27, 41, 73, 119, 120, 121)
- [106] SOLER (Cyril), *Représentations Hiérarchiques de la Visibilité pour le Contrôle de l'Erreur en Simulation de l'Eclairage*, Thèse de doctorat, Université Joseph Fourier, Grenoble, France, décembre 1998. (cité à la page 42)
- [107] SPARROW (E. M.), « Application of Variational Methods to Radiation Heat-Transfer Calculations », *ASME Journal of Heat Transfer*, 82, n° 4, novembre 1960, p. 375–380. (cité à la page 17)
- [108] _____, « A New and Simpler Formulation for Radiative Angle Factors », *ASME Journal of Heat Transfer*, 85, n° 2, mai 1963, p. 81–88. (cité à la page 21)
- [109] SPARROW (E. M.) et CESS (R. D.), *Radiation Heat Transfer*, Washington, Hemisphere Publishing Corporation, 1978. (cité à la page 17)
- [110] STAMMINGER (Marc), SCHEEL (Annette), GRANIER (Xavier), PEREZ-CAZORLA (Frederic), DRETTAKIS (George) et SILLION (François X.), « Efficient Glossy Global Illumination with Interactive Viewing », *Computer Graphics Forum*, 19, n° 1, 2000, p. 13–25. (cité à la page 111)
- [111] STAMMINGER (Marc), SCHIRMACHER (Hartmut), SLUSALLEK (Philipp) et SEIDEL (Hans-Peter), « Getting Rid of Links in Hierarchical Radiosity », *Computer Graphics Journal (Proc. Eurographics '98)*, 17, n° 3, septembre 1998, p. C165–C174. (cité à la page 30)
- [112] STOLLNITZ (Eric J.), DEROSE (Tony D.) et SALESIN (David H.), *Wavelets for Computer Graphics : Theory and Applications*, San Francisco, CA, Morgan Kaufmann, 1996. (cité à la page 28)
- [113] SUDARSKY (Oded), « Exploiting Temporal Coherence in Animation Rendering : A Survey », Rapport Technique CIS 9326, Center for Intelligent Systems, Technion - Israel Institute of Technology, Technion, Israel, novembre 1993. (cité à la page 37)
- [114] TELLER (Seth) et HANRAHAN (Pat), « Global Visibility Algorithms for Illumination Computations », dans *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, p. 239–246, 1993. (cité à la page 18)
- [115] TROUTMAN (Roy) et MAX (Nelson L.), « Radiosity Algorithms Using Higher Order Finite Element Methods », dans *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, p. 209–212, 1993. (cité à la page 28)
- [116] TSINGOS (Nicolas), *Simulation de Champs Sonores de Haute Qualité Pour Des Applications Graphiques Interactives*, Thèse de doctorat, Université Joseph Fourier, Grenoble, France, décembre 1998. (cité à la page 50)
- [117] TSINGOS (Nicolas) et GASCUEL (Jean-Dominique), « A general model for the simulation of room acoustics based on hierarchical radiosity », dans *SIGGRAPH'97 Visual proceedings, technical sketch, Los Angeles, USA*, août 1997. (cité à la page 50)
- [118] WALTER (Bruce), DRETTAKIS (George) et PARKER (Steven), « Interactive Rendering using the Render Cache », dans LISCHINSKI (D.) et LARSON (G.W.), éditeurs, *Rendering techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering)*, volume 10, p. 235–246, New York, NY, juin 1999, Springer-Verlag/Wien. (cité aux pages 38, 39, 41)

- [119] WARD (Gregory J.), A Contrast-Based Scalefactor for Luminance Display, dans HECKBERT (Paul S.), éditeur, *Graphics Gems IV*, p. 415–421, Boston, MA, Academic Press Professional, 1994. (cité à la page 31)
- [120] WILLMOTT (Andrew) et HECKBERT (Paul), « An Empirical Comparison of Progressive and Wavelet Radiosity », dans DORSEY (Julie) et SLUSALLEK (Philipp), éditeurs, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, p. 175–186, New York, NY, 1997, Springer Wien, ISBN 3-211-83001-4. (cité aux pages 29, 119)
- [121] ZATZ (Harold R.), « Galerkin Radiosity : A Higher Order Solution Method for Global Illumination », dans *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, p. 213–220, 1993. (cité aux pages 20, 28)

Résumé

Les méthodes globales de simulation de l'éclairage permettent, à la différence des méthodes locales, d'exprimer l'équilibre énergétique dans les échanges entre différents objets, et donc de simuler précisément les effets subtils d'éclairage dus aux (nombreuses) inter-réflexions. Il est donc naturel de souhaiter les utiliser pour la synthèse réaliste de films d'animation. Plutôt que de résoudre une succession d'équations intégrales tri-dimensionnelles, nous modélisons les échanges lumineux ayant lieu au cours de l'animation sous la forme d'une unique équation intégrale quadri-dimensionnelle. Dans le cas où l'intégralité des mouvements est connu à l'avance, nous proposons une extension de l'algorithme de radiosit  hi rarchique mettant   profit la coh rence temporelle. La radiosit  en chaque point et   chaque instant y est exprim e dans une base de fonctions hi rarchiques d finies sur un maillage produit par un processus de raffinement. L'extension de ce maillage   un espace   quatre dimensions nous permet de calculer des  changes lumineux sur un intervalle de temps fini au lieu d'une date donn e. L'algorithme ainsi d fini permet la simulation globale de l' clairage diffus dans une sc ne anim e, dans un temps largement inf rieur   un calcul image par image, avec une qualit   quivalente. Nous avons d velopp  pour cela de nouveaux oracles de raffinement *ad hoc*, que nous pr sentons ici. Afin de permettre le calcul de sc nes g om triquement complexes, nous pr sentons une nouvelle politique de regroupement hi rarchique des objets adapt e au cas quadri-dimensionnel. Nous pr sentons  galement un algorithme permettant la r duction des discontinuit s temporelles dues aux approximations effectu es lors de la r solution, bas  sur l'emploi de bases de multi-ondelettes. Finalement, nous pr sentons un m canisme d'ordonnancement des calculs et de sauvegarde temporaire sur une m moire de masse permettant de r duire la consommation en m moire vive de l'algorithme.

Abstract

In image synthesis, the global illumination family of methods relies on the expression of the energy equilibrium between objects, and as a consequence allows the precise simulation of subtle lighting effects due to numerous reflections of light, that could not be taken into account by local lighting models. Therefore, it seems logical to try to use them for the synthesis of realistic computer animations. In this document, we suggest to replace the computation of the numerous integral equations corresponding to each frame by a single four-dimensional equation. We propose an extension of the hierarchical radiosity algorithm, in the case where all movement is known a priori, that allows us to benefit from temporal coherence. The radiosity at each point of the scene, for each date of the animation, is expressed in a hierarchical function basis defined over a mesh produced by an automatic refinement process. The extension of this mesh to a four-dimensional space allows the computation of light exchanges over a finite time interval instead of a given fixed date. This algorithm allows the simulation of global light exchanges in an animated scene, faster and with a quality similar to frame by frame computations. For this purpose, we present new dedicated refinement oracles. To allow the computation of geometrically complex scenes, we discuss the extension of clustering approaches to the case of animated scenes. We also introduce a method to reduce the temporal discontinuities in the radiosity function based on the use of higher order basis functions (MultiWavelets) in the time dimension. Finally, we present a mechanism to schedule the space-time hierarchical radiosity computations to allow the use of disk caching and reduce the amount of memory required by our algorithm.