



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

UNIVERSITE DE METZ

THESE

Spécialité : INFORMATIQUE

présentée le 18 janvier 1993
en vue de l'obtention du
grade de docteur de l'université
de METZ

par

Robin VIVIAN

BIBLIOTHEQUE UNIVERSITAIRE - METZ	
N° inv.	1993005S
Cote	S/M3 93/2
Loc	Magasin

Sujet:

**DEFINITION D'UNE APPROCHE ADAPTATIVE:
APPLICATION A LA VISUALISATION
EN CAO**

MM	Yvon	GARDAN	Directeur de thèse
	Bernard	PEROCHE	Rapporteur
	Michel	MERIAUX	Rapporteur
	Alain	MOLINARI	Examineur
	J.Claude	PAUL	Examineur

Ainsi, en connaissant jours après jours mon maître, et en passant nos longues heures de marche en de très longues conversations dont, le cas échéant, je parlerai au fur et à mesure, nous parvînmes au pied du mont où se dressait l'abbaye. Et il est temps, comme jadis nous le fîmes, que mon récit s'approche d'elle: puisse ma main ne point trembler au moment où je m'apprête à dire tout ce qui arriva

Umberto ECO
" le nom de la rose"

A Hélène et Nino pour hier
A Isabelle pour demain

REMERCIEMENTS

La réalisation d'un travail de recherche, à partir d'un sujet aussi bon soit-il, est largement tributaire de l'environnement dans lequel il se situe et de l'encadrement dont il bénéficie. La sagesse et la valeur des gens que j'ai côtoyés tout au long de cette thèse m'a permis de comprendre combien il était important de croire en ce que l'on faisait aussi modeste que soit notre contribution au domaine exploré.

C'est pourquoi je ne peux commencer ce manuscrit sans exprimer toute ma reconnaissance à **Yvon GARDAN**, professeur à l'université de METZ, qui, au-delà de son rôle de directeur de thèse, a su me faire comprendre la nécessité de l'échec et la fragilité de la réussite.

Je tiens à témoigner toute ma gratitude à **M. MERIAUX**, professeur à l'université de LILLE, et **M. PEROCHE**, professeur à l'école des mines de Saint-ETIENNE qui malgré leurs multiples occupations ont bien voulu juger et corriger ce manuscrit.

Je tiens à remercier tout aussi sincèrement **M. MOLINARI**, professeur à l'université de METZ, qui bien que mécanicien dans l'âme, a accepté avec beaucoup de prévenance d'être examinateur sur une thèse informatique.

Je remercie tout spécialement **M. PAUL** architecte et directeur de Centre de Recherche en Architecture et Ingénierie de NANCY, examinateur et qui reste à l'origine de la réalisation de ce travail.

Je tiens aussi à exprimer toute ma gratitude à la société **T.D.I.** de PARIS et plus particulièrement à **M. NICOLAS**, directeur de recherche et développement, pour son soutien financier durant les deux premières années de mon travail.

Je n'oublie aucunement les membres du **L.R.I.M.**, chercheurs, techniciennes (M'an Jocelyne, Martine, Dominique) ainsi que Catherinette, qui bien que tous ne soient pas fréquentables, restent quand même une des grandes satisfactions de ces trois années. Je remercie plus particulièrement, Yannos (Yann LANUEL) qui, en bon disciple, a su m'épauler dans la réalisation de ce travail, à charge de revanche.

Je ne saurais terminer sans rendre un hommage à Isabelle, qui a supporté pendant trois ans les week end passés devant une machine, les moments de repos furtifs, les corrections toujours urgentes. Pour tout ce que tu as su m'apporter dans les moments de doutes et pour tout ce que tu me donneras je te dédie ce travail pour te dire combien je suis fier de partager ta vie.

Table des matières

1. INTRODUCTION.....	5
CH1 ALGORITHMES DE RENDU : METHODES PRINCIPALES	7
1. LA RADIOSITE	7
1.1. Echanges lumineux entre deux surfaces.....	8
1.2. Rapide bilan de la radiosit�.....	10
2. LE TRACE DE RAYONS.....	11
2.1. L'algorithme du trac� de rayons.....	11
2.2. Traitement des faces cach�es	12
2.3. Les rayons primaires	12
2.4. Traitement des r�flexions et r�fractions	13
2.3.1. La r�flexion sp�culaire.....	13
2.3.2. La r�fraction.....	14
2.5. Les rayons d'ombrage	15
2.6. Rapide bilan du trac� de rayons	16
CH2 TRACE DE RAYONS : LES TECHNIQUES D'OPTIMISATION	17
1. INTRODUCTION.....	17
1.1. Optimisation du calcul d'intersection	19
1.1.1. Acc�l�ration du calcul par optimisation de l'algorithme d'intersection rayon-objet	19
1.1.2. Acc�l�ration du calcul par diminution du nombre d'intersections rayon- objet.....	23
1.2. Optimisation par recherche de coh�rence entre les rayons.....	24
1.3. G�n�ralisation de la notion de rayon.....	24
2. LES DECOMPOSITIONS.....	25
2.1. Les subdivisions non uniformes	25
2.1.1. Un exemple de d�composition non uniforme avec un arbre octal: l'approche de Glassner.....	28
2.1.2. Un exemple de d�composition non uniforme avec cellules: La d�composition d'Arnaldi, Priol, Bouatouch.....	30
2.1.3. La d�composition non uniforme avec macro-r�gions: l'approche de Devillers	32
2.2. Les subdivisions uniformes.....	33
2.2.1. D�compositions uniformes: l'approche de Fujimoto, Tanaka et Iwata.....	34
3. DIMINUTION DU NOMBRE DE RAYONS CALCULES.....	35
3.1. Interpolation: l'approche d'Akimoto.....	36
3.2. Exemple de technique directionnelle: cube directionnel	37
3.3. Exemple de technique directionnelle: La coh�rence des rayons.....	39
4. LA GENERALISATION DE RAYONS.....	40
4.1. G�n�ralisation de rayons: l'approche d'Amanatides: Ray-tracing avec c�nes	40
4.2. Exemple de g�n�ralisation de rayons: l'approche d'Heckbert, Hanrahan : le beam tracing.....	41
5. OPTIMISATION POUR ARBRE CSG.....	43
5.1. Utilisation des arbres C.S.G : L'approche de Wyvill et Kunii	44
6. COMPARATIF DES METHODES	45
7. LES OPTIMISATIONS FUTURES.....	46

CH3 DEFINITION D'UNE APPROCHE ADAPTATIVE.....	47
1. INTRODUCTION.....	47
2. LIMITES DES DECOMPOSITIONS CLASSIQUES.....	51
3. DEFINITION D'UNE DEFINITION ADAPTATIVE.....	53
4. ADAPTABILITE A LA POSITION ET A LA FORME DES OBJETS: application aux rayons primaires.....	54
5. ADAPTABILITE A LA POSITION ET LA FORME DES OBJETS: traitement des rayons secondaires et d'ombrage.....	59
5.1. Les rayons secondaires.....	59
5.1.1. Les rayons réfractés.....	59
5.1.2. Les rayons réfléchis.....	62
5.2. Les rayons d'ombrage.....	63
5.2.1. L'éclairage direct.....	63
5.2.2. L'éclairage indirect.....	65
5.3. Méthode générale.....	67
6. ADAPTABILITE AU CONTENU DE LA SCENE.....	68
6.1. Les tests d'arrêt terminaux.....	69
6.2. Les tests d'orientation.....	69
6.2.1. Les tests d'orientation : le taux d'occupation.....	69
6.2.1.1. définition du test.....	69
6.2.1.2. Application du test.....	70
6.2.2. les tests d'orientation : le nombre d'objets maximal.....	72
6.2.3. les tests d'orientation : égalité de contenu entre les sous-fenêtres.....	72
7. ADAPTABILITE DES TESTS ET DES VALEURS A L'UTILISATEUR.....	74
7.1. Pour les images de petite taille.....	75
7.2. Pour les images de grande taille.....	76
7.2.1. Le test du nombre d'objets maximum.....	76
7.2.2. Le taux d'occupation.....	77
8. ADAPATABILITE DES ALGORITHMES AUX OBJETS.....	77
8.1. Adaptabilité aux caractéristiques optiques des objets.....	78
8.2. Adaptabilité aux caractéristiques géométriques des objets.....	80
9. CONCLUSION.....	82
CH4 VALIDATION DE LA METHODE.....	84
PRESENTATION DES RESULTATS.....	84
1. INTRODUCTION.....	84
2. PRESENTATION DES CONDITIONS DE TEST.....	84
3. PRESENTATION DE LA METHODE GENERALE.....	86
3.1. Optimisation des rayons primaires et d'ombrages.....	87
3.1.1. Les caractéristiques de la décomposition.....	87
3.1.2. Les temps de calcul.....	88
3.1.3. Interprétation des résultats.....	88
3.1.4. Conclusion.....	90
3.2. Optimisation des rayons primaires, d'ombrages et réfléchis.....	91
3.2.1. Les caractéristiques de la décomposition.....	91
3.2.2. Les temps de calcul.....	92
3.2.3. Interprétation des résultats.....	93
3.2.3.1. Scènes ne contenant que des objets réfléchissants de révolution.....	93
3.2.3.2. Scènes contenant des objets réfléchissants de type plan et des objets de révolution.....	94
3.2.3.3. Scènes ne contenant que des objets réfléchissants de type plan.....	96
3.2.4. Conclusion.....	97
3.3. Optimisation des rayons primaires, d'ombrage et réfractés.....	98
3.3.1. Les caractéristiques de la décomposition.....	98
3.3.2. les temps de calcul.....	99
3.3.3. Interprétation des résultats.....	100
Analyse des valeurs.....	100
3.3.4. Conclusion.....	101

4. INFLUENCE DES TESTS D'ARRET	102
4.1. Recherche des valeurs à attribuer aux paramètres	102
4.1.1. Variation du nombre d'objets par sous-fenêtre	102
4.1.2. Variation de la valeur du taux d'occupation	103
4.1.3. Variation de la taille de la sous-fenêtre	104
4.1.4. Conclusion	106
4.2. Choix des tests d'arrêt.....	106
4.2.1. Scènes occupant toute la fenêtre.....	107
4.2.2. Scènes ne contenant qu'un objet.....	107
4.2.3. Scènes où les objets sont séparés.....	107
4.2.4. Conclusion	108
5. CONCLUSION.....	108
CONCLUSION.....	110
BIBLIOGRAPHIE	112
ANNEXE 1.....	119
ANNEXE 2.....	125
ANNEXE 3.....	131

1. INTRODUCTION

Le XXI^{ème} siècle sera celui de l'image ou ne sera pas !

Depuis l'invention des frères Lumière l'utilisation de l'image n'a cessé de croître pour être de nos jours, un support indispensable à une communication efficace. Marginale il n'y a encore que quelques années elle n'est en fait utilisée que dans certains milieux industriels uniquement pour des raisons de coût de fabrication. En effet, obtenir une image un tant soit peu réaliste nécessitait il y a 10 ans des jours de calcul sur d'imposantes et coûteuses machines. L'image de synthèse se démocratise au début des années 80 lorsqu'elle prend une coloration artistique grand public. Les spots publicitaires et des films Américains à gros budgets vont devenir des consommateurs d'une technologie qui devient de plus en plus abordable. Cependant l'industrie reste le plus gros consommateur d'image avec plus de 50% du marché mondial.

Cette évolution du marché va être due à trois phénomènes concourants. Tout d'abord l'évolution technologique dédiée à l'image de synthèse, la baisse des prix des machines liée à leur accroissement de puissance va permettre au plus grand nombre de réaliser, peut être pas encore simplement, des images. Ensuite, l'évolution des algorithmes qui permet de modéliser des phénomènes lumineux de plus en plus complexes, la course au réalisme est lancée. Le troisième et dernier phénomène est l'ouverture vers de nouveaux secteurs comme l'architecture où le crayon et les pastels se dressaient en défenseurs de traditions désuètes.

Le processus de fabrication d'une image de synthèse se décompose en deux étapes distinctes. La première est la modélisation des composantes géométriques de la scène à savoir la construction des objets, leur état de surface, leur mouvement. La seconde consiste à définir un éclairage de la scène en simulant des effets lumineux comme la réflexion ou la réfraction. Il existe cependant des liens qui unissent ces deux étapes ce sont d'une part l'exactitude des représentations, d'autre part la vitesse de calcul.

Les problèmes abordés dans cette thèse sont avant tout l'amélioration des temps de calcul de l'algorithme de tracé de rayons par une analyse adaptative du contenu d'une scène et l'application sélective des algorithmes d'optimisation.

La méthode de décomposition adaptative que nous développons repose sur une décomposition récursive de type quadtree. Le premier inconvénient de cette méthode est une subdivision "aveugle" de l'espace de travail provoquant des échantillonnages abusifs souvent inutiles. Le second est que des décompositions surfaciques ne s'appliquent que très difficilement aux rayons secondaires et aux rayons d'ombrage.

Nous apportons une solution à ces deux problèmes en définissant une méthode de décomposition qui repose sur trois principes essentiels:

- Le premier est la définition d'une décomposition qui s'adapte en fonction du contenu de la scène. On évalue à chaque étape de la subdivision la nécessité de poursuivre ou non la décomposition en fonction de critères différents de ceux utilisés habituellement dans les subdivisions récursives comme par exemple la répartition spatiale ou encore le taux d'occupation.

- Le second est la création d'un faisceau de visualisation (s'apparentant au beam tracing) que l'on essaye de propager en fonction des caractéristiques optiques et géométriques des objets. On effectue, à l'intérieur de ces pyramides de visualisation une localisation des objets par une projection perspective afin de limiter l'application de l'algorithme de tracé de rayons.

- Le troisième est l'adaptabilité des principes d'optimisation à la géométrie des objets. La contrainte forte d'une méthode proche du beam tracing est de ne s'appliquer qu'à des objets approchés par des faces planes. En effet, si un objet spéculaire est de révolution on ne peut propager une pyramide de visualisation. Afin de ne pas limiter les scènes créées à un ensemble d'objets parallélépipédiques nous modifions les algorithmes d'optimisation en fonction de l'objet traité.

Le chapitre 1 est une présentation rapide des deux principaux algorithmes de visualisation à savoir l'algorithme de radiosité et celui du tracé de rayons. Nous effectuons au chapitre 2 une description la plus complète possible de toutes les grandes techniques d'optimisation du tracé de rayons avant d'exposer au chapitre 3 la méthode de décomposition adaptative. Le chapitre 4 sera consacré à la validation de la méthode par une présentation des tests effectués sur un ensemble de scènes types. Les différents modèles utilisés ainsi que les images calculées sont présentés dans les différentes annexes.

Chapitre 1

ALGORITHMES DE RENDU : METHODES PRINCIPALES

Nous présentons dans ce chapitre les deux méthodes de rendu réaliste les plus utilisées à savoir la radiosité et le tracé de rayons. Ce sont en fait les deux seules méthodes qui permettent de prendre en compte l'ensemble des phénomènes optiques conventionnels comme les transparences, les réflexions. Bien que ces méthodes soient généralistes, elles définissent des limites aux phénomènes modélisés. Pratiquement, elles imposent un choix entre les différents types de réflexion et de réfraction.

1. LA RADIOSITE

La radiosité est proposée pour la première fois en image de synthèse par Goral dès 1984 [GOR 84]. Elle est fondée sur le principe de transfert radiatif de chaleur mis en évidence en physique dès 1961 par Gebhart [GEB 61]. Elle permet de prendre en compte tous les problèmes d'éclairage diffus. Les objets composant la scène sont facettisés pour que l'émission totale de lumière par unité de surface soit supposée constante. On notera B_j la quantité totale d'énergie émise par unité de surface et unité de temps en tout point de la facette. On a :

$$B_j = \int_{\Omega} i \cos \theta \, d\omega = i\pi$$

i est l'intensité lumineuse émise. Elle est indépendante de la direction d'émission, les surfaces étant principalement de type lambertien.

Ω est le demi-espace situé au dessus de la facette.

θ est l'angle de réflexion avec la normale à la face.

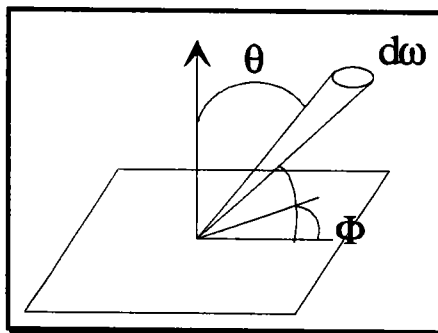


figure 1: puissance réfléchie sur une surface lambertienne

1.1. Echanges lumineux entre deux surfaces

Le principe de la radiativité est de calculer les échanges lumineux entre toutes les surfaces en les étudiant deux à deux.

Soient i et j deux facettes et dA_i , dA_j leur élément de surface.

La quantité d'énergie allant de A_i et arrivant sur A_j est :

$$d^2 P(dA_i, dA_j) = \frac{B_i}{\pi} dA_i \cos \theta_i \frac{dA_j \cos \theta_j}{r^2}$$

L'énergie envoyée par dA_i sur l'ensemble de la facette j s'obtient en intégrant sur la facette j :

$$dP(dA_i, dA_j) = \frac{B_i}{\pi} dA_i \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{r^2} dA_j$$

L'énergie envoyée par toute la facette i vers la facette j est :

$$P(A_i, A_j) = \frac{B_i}{\pi} \int_{A_j} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{r^2} dA_j dA_i$$

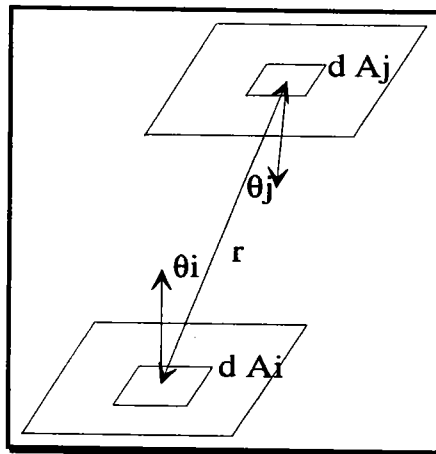


figure 2: transfert d'énergie entre deux surfaces

Si l'on appelle H_j le flux arrivant sur la facette j on peut écrire :

$$H_j = \frac{1}{A_j} \sum_{i \neq j} \frac{B_i}{\pi} \int_{A_j} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{r^2} dA_i dA_j$$

Le flux total émis par une facette est la somme de deux composantes à savoir:

- son émission propre d'énergie
- la réflexion par la facette du flux incident.

Pour un éclairage de type diffus nous avons donc $B_j = E_j + \rho_j H_j$

E_j représente le flux émis par la facette j

Dans leur article Goral *et al.* définissent un calcul simplifié du flux d'énergie. Ils introduisent une valeur appelée *Facteur de forme* qui détermine la forme et la position des objets les uns par rapport aux autres. Le calcul du flux lumineux devient :

$$H_i = \sum_{i \neq j} F_{ij} B_j \text{ avec } F_{ij} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_j dA_i$$

Le calcul du facteur est le point le plus important de la radiosité. De son exactitude vont dépendre les performances et la précision de l'algorithme. Le problème du calcul du facteur de forme est de résoudre une intégrale double. Une des premières solutions proposée par Cohen *et al.* [COH 85] est d'approcher le résultat de cette intégrale double par un hémicube (demi-cube) placé sur le centre de l'élément étudié. Les cinq faces visibles sont décomposées en grille .

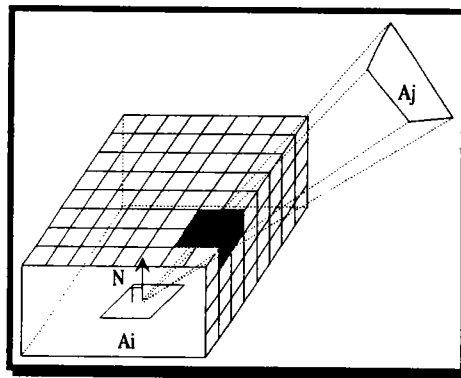


figure 3: l'hémicube

La partie de surface A_j visible par la surface A_i est trouvée en additionnant l'ensemble des pixels intersectés par la projection de A_j . On obtient pour le facteur de forme :

$$F_{ij} = \sum_{q \in Hc} \Delta F_q$$

où q est un pixel de l'hémicube et ΔF_q le facteur de forme associé.

Le principal handicap de la méthode proposée par Goral était que la complexité de l'algorithme en $O(n^2)$ limitait considérablement les performances. Une nouvelle méthode par raffinement progressif de l'éclairage est proposée en 1988 par Cohen [COH 88]. Elle consiste à traiter les faces dans l'ordre croissant de leur radiosité. On calcule la contribution de la face sélectionnée à l'ensemble des autres faces. On peut réaliser alors un premier affichage de la scène bien que celle-ci n'ait qu'un éclairage incomplet. On peut ensuite calculer l'apport des autres faces jusqu'à ce qu'un certain équilibre lumineux soit atteint.

Initialement l'algorithme de radiosité n'était capable de traiter que des surfaces diffuses. L'intérêt principal est que l'éclairage de scène n'est pas à recalculer tant qu'un objet de la scène n'est pas déplacé. Les échanges lumineux sont indépendants de la position de l'observateur. C'est Wallace *et al.* [WAL 87] qui les premiers ont proposé une méthode de calcul d'éclairage intégrant la radiosité et le tracé de rayons pour modéliser l'ensemble des surfaces existantes. Ils décomposent la réflexion de la lumière en quatre méthodes correspondant à tous les cas possibles d'échange entre surfaces. Les quatre transferts possibles sont:

diffus → *diffus*
spéculaire → *diffus*
diffus → *spéculaire*
spéculaire → *spéculaire*

Les deux premiers types d'échange sont correctement traités par un algorithme de radiosité classique, les deux derniers le sont par le tracé de rayons lors d'une seconde *passé*. La méthode perd un peu de son intérêt puisque la seconde *passé* spéculaire nécessite un nouveau calcul partiel de l'éclairage même pour un déplacement de l'observateur.

En 1989 Sillion [SIL 89] propose une méthode qui permet de réaliser les deux calculs en une seule *passé* et propose une nouvelle méthode d'échantillonnage pour la détermination du facteur de forme.

1.2. Rapide bilan de la radiosité

Nous avons succinctement présenté les principes de la radiosité. Nous pouvons terminer ce rapide exposé de la méthode en dressant une liste des points forts et des limites du modèle d'éclairage.

POUR

- * Méthode d'éclairage globale
- * Permet le traitement des surfaces diffuses
- * Traitement des ombres douces et pénombres
- * Méthode incrémentale
- * Indépendante de la position observateur (pour la partie diffuse)

CONTRE

- * Combinaison obligatoire avec un tracé de rayons pour traiter les surfaces spéculaires,
- * Complexité des algorithmes mis en oeuvre
- * Temps de traitement encore important
- * Approximation importante sur le calcul des facteurs de forme donc de l'exactitude du rendu
- * Obligation de facettiser les objets pour simplifier les calculs

2. LE TRACE DE RAYONS

Le premier algorithme de tracé de rayons a été proposé par Appel dès 1968 [APP 68] pour déterminer la visibilité des faces. Il a fallu attendre plus de dix ans pour que Kay [KAY 79] et surtout Whitted [WHI 80] proposent une application du tracé de rayons à l'éclairage global d'une scène. On voit apparaître pour la première fois des images combinant des effets complexes de transparence et de réflexion. Il est nécessaire de préciser que la méthode du tracé de rayons est empirique et ne permet qu'une interprétation très incertaine des lois physiques.

2.1. L'algorithme du tracé de rayons

Le tracé de rayons consiste à simuler le trajet d'un rayon lumineux. Dans la méthode de base les rayons sont considérés comme n'ayant aucune épaisseur. Pour des raisons de faisabilité, le trajet du rayon est le plus souvent pris à rebours. Toutes les lois de propagation se situent dans le cadre d'une optique purement géométrique, c'est-à-dire que les rayons se propagent de manière rectiligne.

Avec ces restrictions le principe de base de l'algorithme de tracé de rayons s'énonce très simplement.

Pour un ensemble d'objets constituant une scène, pour un point origine O et une demi-droite D passant par l'origine, trouver parmi l'ensemble des points d'intersection celui qui est le plus proche de O et à quel objet appartient ce point ?

Il faut pour répondre à cette question être en mesure de calculer un point d'intersection entre une droite et les objets constituant la scène. Il existe pour cela de nombreux algorithmes de calcul d'intersection adaptés aux différentes forme de modélisation des objets (faces planes, csg, Nurbs,...).

La généralisation de ce calcul à un ensemble de rayons issus de l'oeil et passant par les pixels d'un écran fictif, définit l'algorithme de tracé de rayons.

2.2. Traitement des faces cachées

On remarque que le principe de détection du point d'intersection le plus proche traite naturellement le problème de l'élimination des faces cachées.

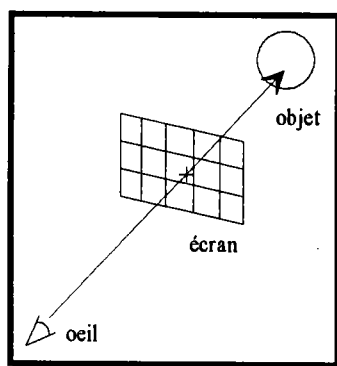


figure 1: détermination des faces visibles

Une fois ce point calculé, il convient de lui associer une modèle d'éclairage permettant de déterminer sa couleur.

2.3. Les rayons primaires

Le rayon primaire est défini comme étant une droite ayant comme origine l'oeil et passant par le centre d'un pixel de l'écran. Il existe donc autant de rayons primaires qu'il y a de pixels sur l'image à calculer. A chacun de ces rayons primaires on applique le principe de détection d'un point d'intersection. Pour les rayons qui ont une intersection effective avec un des objets de la scène, on calcule ensuite l'éclairage du point.

2.4. Traitement des réflexions et réfractions

C'est Whitted [WHI 80] qui le premier a l'idée d'utiliser le tracé de rayons pour calculer l'éclairage d'une scène. Une fois qu'un objet est identifié comme étant visible, il applique au point d'intersection un modèle d'éclairage faisant intervenir une contribution de spécularité et/ou de transparence. La notion de rayon fin ne permet de calculer qu'un rayon réfléchi et/ou réfracté en utilisant les lois de Descartes définies en optique géométrique. L'analyse des rayons étant récursive, il est nécessaire de limiter la profondeur de l'étude.

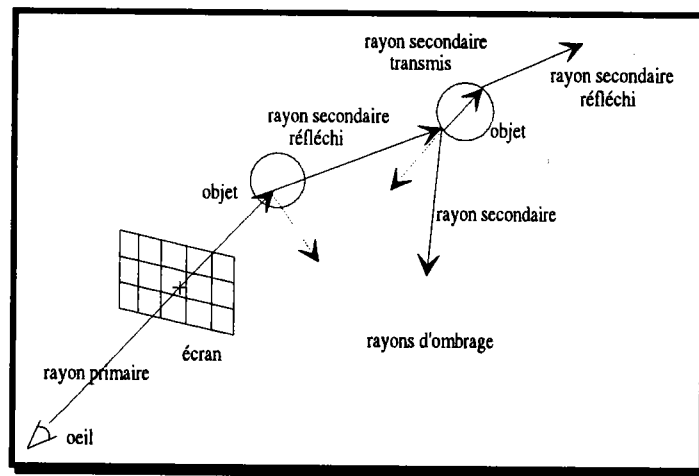


figure 2: tracé de rayons réfléchis et transmis

La couleur finale est trouvée en additionnant à rebours les couleurs des différents objets intersectés par la suite de rayons.

2.3.1. La réflexion spéculaire

La réflexion spéculaire permet de modéliser les miroirs. Elle est en fait le seul type de réflexion qu'un algorithme de tracé de rayons traite correctement. La lumière qui arrive sur la surface est réémise dans une direction de réflexion géométrique. Pour permettre la modélisation de cet effet il faut introduire un coefficient de réflexion du matériau. Le rayon est réfléchi symétriquement par rapport à la normale au point d'intersection. Le rayon incident est appelé rayon secondaire.

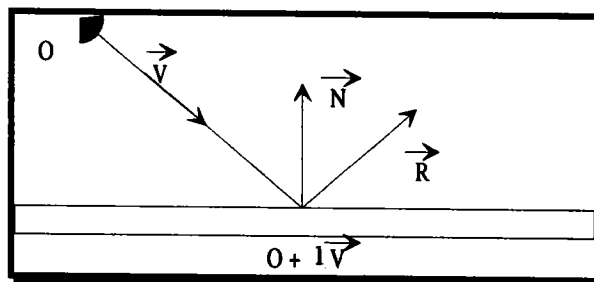


figure 3: direction de réflexion

λ est tel que $P = O + \lambda \vec{v}$

On obtient la direction de réflexion de \vec{r} par :

$$\vec{r} = \vec{v} - 2 \vec{n} \cdot \vec{v} \vec{n}$$

2.3.2. La réfraction

Dans le cas d'objet transparent, la lumière reçue est réémise dans la direction réfractée. La direction du rayon secondaire sera dépendante de l'indice de réfraction du milieu traversé. La direction du rayon réfracté est déterminé par la loi de Descartes:

$$n_1 \sin i_1 = n_2 \sin i_2$$

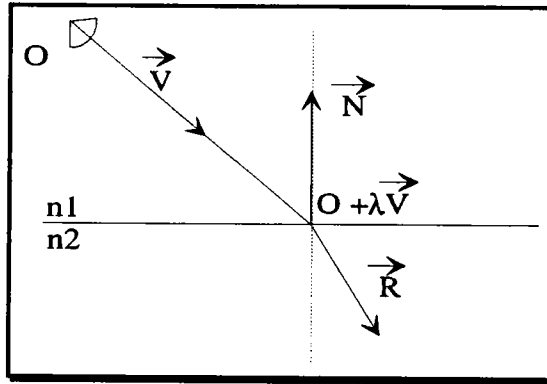


figure 4: Direction de réfraction

Pour trouver la direction de réfraction on décompose \vec{v} et \vec{r} selon la normale et un vecteur \vec{t} tangent à l'objet.

$$\begin{aligned} \vec{v} &= v_n \vec{n} + v_t \vec{t} \\ \vec{r} &= r_n \vec{n} + r_t \vec{t} \end{aligned}$$

On cherche à exprimer \vec{r} , la seule valeur importante, sous la forme $\vec{v} + \mu \vec{n}$ en posant $r_t = n_t$, on a alors :

$$\begin{aligned} \sin i_1 &= \frac{\vec{v} \cdot \vec{t}}{\|\vec{v}\|} = \frac{v_t}{\|\vec{v}\|} \\ \sin i_2 &= \frac{\vec{r} \cdot \vec{t}}{\|\vec{r}\|} = \frac{v_t}{\|\vec{r}\|} \\ \frac{n_2}{n_1} &= \frac{\sin i_1}{\sin i_2} = \frac{\|\vec{r}\|}{\|\vec{v}\|} \\ \left(\frac{n_2}{n_1}\right)^2 &= \frac{r_t^2 + r_n^2}{v_t^2 + v_n^2} = \frac{v_t^2 + r_n^2}{v_t^2 + v_n^2} \end{aligned}$$

On peut calculer r_n et en déduire \vec{r} .

$$r_n^2 = \left(\frac{n_2}{n_1}\right)^2 \|\vec{v}\|^2 - v_t^2$$

En reportant $v_t^2 = \|\vec{v}\|^2 - (\vec{n} \cdot \vec{v})^2$ on a :

$$r_n^2 = (\vec{n} \cdot \vec{v})^2 - \|\vec{v}\|^2 \left[1 - \left(\frac{n_2}{n_1}\right)^2 \right]$$

Si cette valeur devient négative le rayon lumineux n'est plus réfracté mais il devient réfléchi sinon \vec{r} est égal à :

$$\vec{r} = \vec{v} - \left(\vec{n} \cdot \vec{v} + \sqrt{(\vec{n} \cdot \vec{v})^2 - \|\vec{v}\|^2 \left[1 - \left(\frac{n_2}{n_1}\right)^2 \right]} \right) \vec{n}$$

2.5. Les rayons d'ombrage

Nous avons présenté jusqu'à présent deux types de rayon, les rayons primaires et les rayons secondaires (réfléchis et réfractés). Pour calculer l'intensité lumineuse en chaque point de la scène il faut calculer la contribution de chaque source lumineuse. Pour cela on lance un rayon vers chacune de ces sources "le rayon d'ombrage" et nous regardons si ce rayon coupe un des objets. Si une intersection existe, le point étudié se trouve dans l'ombre d'un objet. Dans ce cas, on modifie les paramètres lumineux du point.

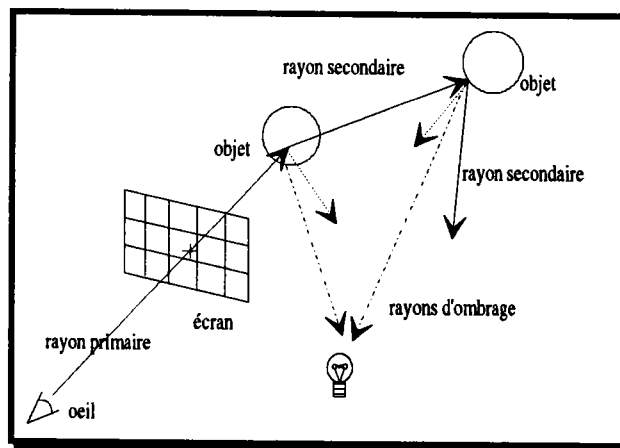


figure 5: les rayons d'ombrage

2.6. Rapide bilan du tracé de rayons

Comme pour l'algorithme de radiosit  nous allons terminer cette pr sentation de l'algorithme de trac  de rayons en r alisant un rapide bilan des principaux points forts et inconv nients de l'algorithme.

POUR

- * Simplicit  de mise en oeuvre de l'algorithme de base.
- * S'adapte   tous les mod les (CSG, faces planes, NURBS,...).
- * M thode globale qui permet de prendre en compte les surfaces r flechissantes et transparentes.
- * Prise en compte des ombres port es.

CONTRE

- * Le calcul de l'image est d pendant du point de vue.
- * Les temps de calcul sont importants pour la m thode de base.
- * Ne prend pas en compte les lumi res diffuses.
- * G n re d'importants probl mes d'aliassage dus au concept de rayons fins (d' chantillonnage ponctuelle de la sc ne).
- * Il est limit  dans le type des sources lumineuses repr sentables (essentiellement ponctuelles)

Les principaux d fauts du trac  de rayons sont encore aujourd'hui l'aliassage et surtout des temps de calcul importants. Depuis Appel, l'ensemble des travaux sur le trac  de rayons a consist    d finir un nouveau concept de rayon et   diminuer les co t de traitement. Nous allons pr senter au chapitre 2 l'ensemble des techniques qui permettent de palier ces inconv nient, en reprenant plus en d tail des plus importantes d'entre elles.

Chapitre 2

TRACE DE RAYONS : LES TECHNIQUES D'OPTIMISATION

1. INTRODUCTION

L'un des handicaps du tracé de rayons, décrit la première fois par Appel en 68 [APP 68] est qu'il est relativement coûteux en temps de calcul. Il faut, encore aujourd'hui, plusieurs heures de calcul sur des machines puissantes pour obtenir des images contenant un grand nombre d'objets avec des effets de rendu réaliste. L'accélération des algorithmes reste une préoccupation importante des développeurs de logiciels de C.A.O et plus particulièrement les algorithmes dédiés à l'image de synthèse. La recherche de gain de temps a dépassé la simple performance intellectuelle des chercheurs, elle est aussi imposée par des contraintes économiques et de confort. Même si l'on sait que les puissances des machines seront multipliées par 10 dans les années à venir, notamment avec les machines parallèles, leur vitesse de travail ne permettra pas seule d'arriver au but inavoué de tout concepteur de logiciel d'images de synthèse à savoir: être capable de faire de l'animation temps réel de scènes complexes. C'est pourquoi l'on voit toujours se développer de nouvelles techniques d'optimisation de tracé de rayons (ray-tracing).

La figure 1 résume les différentes stratégies mises en oeuvre.

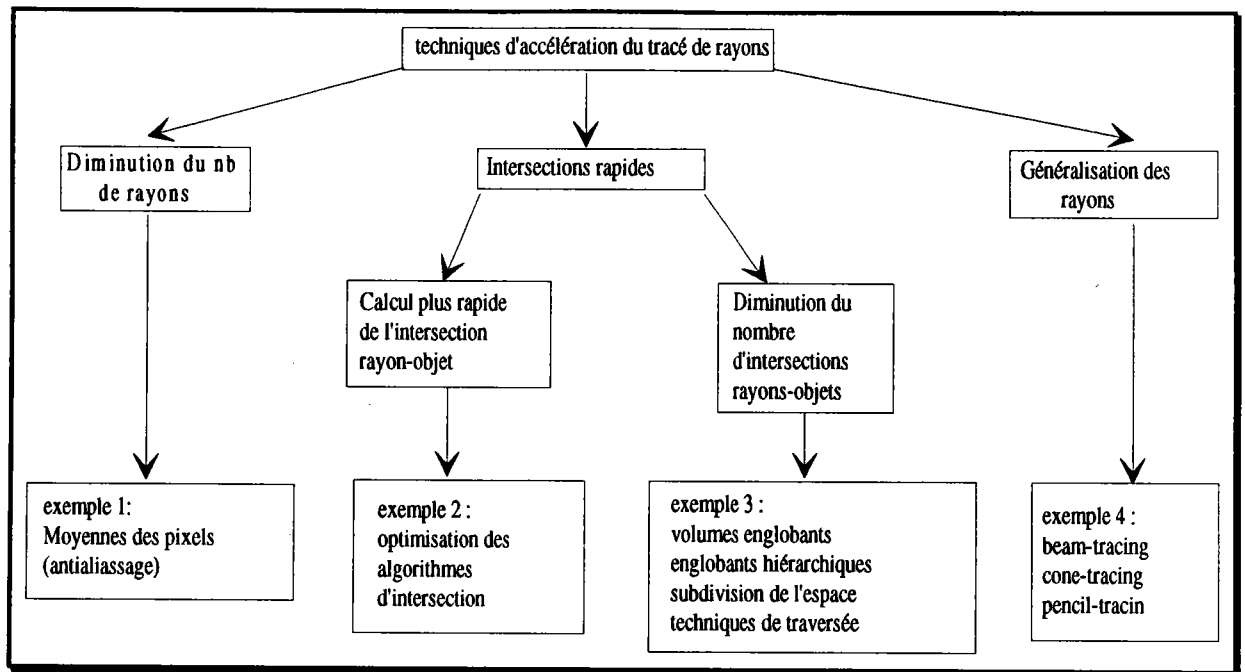


figure 1 : classification des techniques d'optimisation

On remarque que n'apparaissent pas dans cette classification les optimisations dues aux machines parallèles. On s'est intéressé essentiellement aux optimisations algorithmiques se contentant simplement d'expliquer sommairement le principe des machines parallèles et de citer certains travaux. Le principe est de multiplier le nombre de micro-processeurs ceux-ci ne traitant qu'un nombre très restreint de pixels écran. La machine idéale étant une machine possédant un processeur par pixel. Cette approche semble très séduisante notamment pour le traitement des faces cachées. Elle présente par contre un certain nombre de lacunes dès que l'on essaye d'introduire de la cohérence dans l'analyse qui est faite de la scène. La parallélisation massive du traitement se fait au détriment d'un certain nombre d'informations et de renseignements très utiles sur les liaisons de dépendance entre objets (faces cachées, ombrage, zones d'influence), les liaisons mécaniques (notions de tangence). Le concept de parallélisation semble encore moins performant dès que l'on aborde l'animation. Il ne semble pas permettre dans l'état actuel des travaux une réutilisation d'informations d'une image à l'autre puisqu'il n'y a pas une connaissance de la scène d'une manière globale et "intelligente". En fait, il semble manquer aux machines parallèles une méta-couche capable de conserver toutes les informations nécessaires à une bonne compréhension du contenu d'une scène d'un point de vue conceptuel. On citera pour mémoire les travaux de Dippé [DIP 84] et Maurel [MAU 91].

Contrairement à cette approche, la grande majorité des méthodes que l'on présentera et qui peuvent s'intégrer dans une des sous-parties de la figure 1 tentent de minimiser les calculs en simplifiant la scène par une décomposition ou une déduction des informations des analyses précédentes mais en conservant toujours des informations géométriques définissant la scène.

Dans les techniques d'accélération du tracé de rayons, on a distingué trois grandes familles qui correspondent à trois stratégies très différentes. Il est évident que la frontière entre ces différentes approches n'est pas toujours aussi évidente que sur ce schéma.

1.1. Optimisation du calcul d'intersection

La première famille que l'on a distingué est celle que l'on a intitulé "intersection rapide". On a regroupé dans cette catégorie les algorithmes cherchant à optimiser le calcul d'intersection. On a distingué deux grandes tendances:

- l'accélération du calcul par optimisation de l'algorithme d'intersection rayon-objet
- l'accélération du calcul par diminution du nombre de calculs d'intersections

1.1.1. Accélération du calcul par optimisation de l'algorithme d'intersection rayon-objet.

On regroupe dans cet ensemble tous les algorithmes optimisant le calcul de tracé de rayons par une amélioration de la partie mathématique de l'intersection. On n'a indiqué dans cette sous-partie que la méthode des englobants [WHI 80], mais on aurait pu citer tous les travaux effectués sur l'optimisation des intersections rayon-carreau nurbs [CAT 74][SED 84][SAH 89][KAJ 82][KAY 86][CLE 88][ROT 82][SWE 86] qui sont des méthodes calculatoires lourdes à mettre en oeuvre et où il est courant de résoudre des systèmes d'équations de degré supérieur à 18.

La technique des volumes englobants reste l'une des optimisations les plus générales et les plus simples à mettre en oeuvre. Elle est générale dans le sens où elle s'applique aussi bien aux rayons primaires qu'aux rayons secondaires ainsi qu'aux rayons d'ombrage. Un volume englobant est un volume simple, généralement une sphère ou un parallépipède rectangle, approchant la forme d'un objet. Cette approximation permet d'affirmer qu'un rayon coupe potentiellement un objet uniquement s'il coupe dans un premier temps son volume englobant. La recherche d'intersections entre une droite (représentative d'un rayon) et une sphère ou un parallépipède, plus simple que la recherche d'intersections entre une droite et une surface quelconque, permet de diminuer considérablement les calculs inutiles.

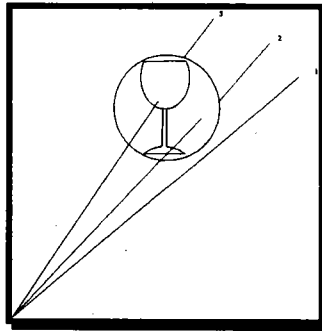


figure 2: intersection avec un englobant

La figure 2 présente très succinctement le principe de la méthode en mettant en évidence les trois cas d'intersections possibles. On remarque que le rayon 1 ne coupe pas la sphère englobante et ne pourra donc en aucun cas couper l'objet. L'intersection de l'englobant est une condition nécessaire mais non suffisante pour que le rayon coupe effectivement l'objet. Ce cas de figure est illustré sur l'exemple ci-dessus par le rayon 2. Le rayon 3 coupant à la fois la sphère englobante et l'objet sera le seul retenu pour afficher l'image de l'objet.

L'exemple ci-dessus présente bien le problème des englobants qui est qu'un rayon coupant un englobant ne coupe pas forcément l'objet (cas du rayon 2). S'est posée alors la question du choix du meilleur englobant. Un bon englobant se caractérise par:

- sa capacité à bien épouser la forme de l'objet
- la nécessité que le calcul rayon/englobant reste simple.

Il n'existe pas d'englobant universel. Certains seront "moins pires" que d'autres en fonction de la forme de l'objet.

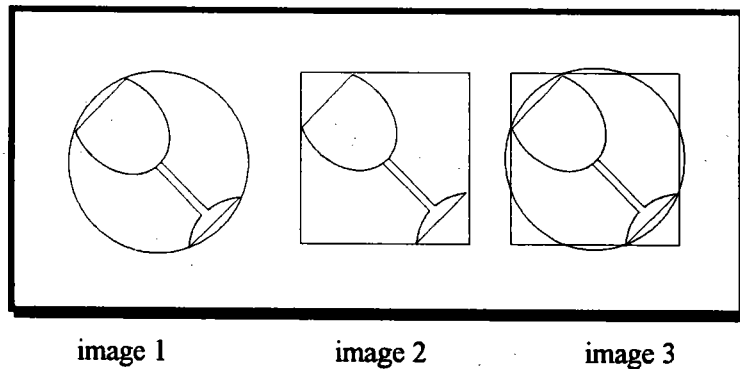


figure 3: englobants usuels

La figure 3 présente les trois types d'englobants les plus utilisés.

* L'image 1 montre un englobant sphérique. Si la sphère reste une excellente approximation pour tout un ensemble d'objets, on remarque qu'il n'épouse que très grossièrement la forme du verre. Le calcul de l'englobant peut être encore optimisé.

* L'image 2 propose un englobant parallépipédique avec des faces parallèles aux bords de l'écran. C'est l'une des approximations la plus utilisée dans les algorithmes d'optimisation de rendu. Elle a l'immense avantage d'être plus performante que les sphères sur des objets présentant d'importantes variations dans deux dimensions sur trois (fig 4).

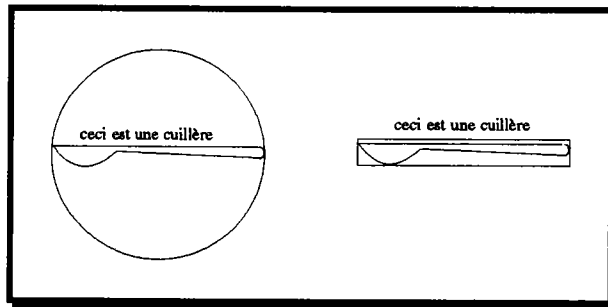


figure 4: différence entre englobants

* L'image 3 représente une optimisation réalisée à partir d'un englobant sphérique et d'un englobant parallépipédique. L'objet est approché par l'intersection des deux types d'englobants, ce qui permet d'obtenir une meilleure approximation de l'enveloppe de l'objet. Lorsque l'objet est allongé (fig 4), l'englobant résultant est proche de l'englobant parallépipédique qui approche bien mieux l'objet qu'un englobant sphérique et inversement si l'objet est relativement compact. Quand l'objet possède une forme indéterminée, l'intersection des deux volumes est toujours plus performante que chacun des englobants pris séparément.

A partir de ces deux englobants de base, on a recherché des compromis permettant de minimiser, pour tous les types d'objets, le calcul d'intersection rayon, objet.

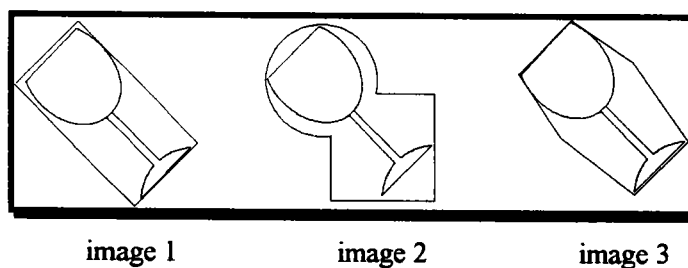


figure 5: autres types d'englobants

* L'image 1 présente un englobant parallépipédique avec des bords non parallèles à l'écran. Cette approximation est bien meilleure, d'un point de vue géométrique, qu'un englobant parallépipédique avec bords parallèles aux axes, le problème est qu'il engendre un surcoût dans le calcul d'intersections.

* L'image 2 présente un autre compromis entre les volumes sphériques et parallélépipédiques. On a vu, figure 3, que l'on pouvait optimiser les calculs en prenant l'intersection entre englobants; Ici l'idée est de prendre l'union de deux sous-volumes: un sphérique approchant mieux la partie haute du verre et un parallélépipédique pour la partie du pied.

* Le dernier exemple, illustré par l'image 3, montre un volume englobant obtenu à l'union de deux volumes parallélépipédiques décalés d'un angle β .

Rubin et Whitted proposent en 80 [WHI 80] une hiérarchisation de la notion des volumes englobants. Ils définissent des volumes englobants par objet puis par groupe d'objets. Par la suite l'idée a été reprise et adaptée par Weghost en 84 et Kay en 86 [WEG 84] [KAY 86]

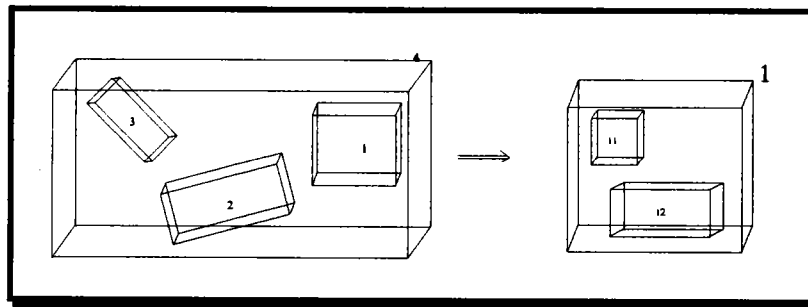


figure 6: englobants hiérarchiques

Pour un rayon coupant la boîte 4 on regarde si un des sous-volumes englobant est intersecté (1,2 ou 3). Si l'englobant 1 a une intersection avec le rayon, on regarde si un des sous-englobants de 1 (11,12) est intersecté par le rayon et ainsi de suite jusqu'à trouver une intersection effective avec un objet ou que le rayon ne coupe aucun des englobants.

Les techniques d'utilisation des englobants ont permis de diminuer de manière très sensible les temps de calcul des algorithmes de rendu comme le Ray-Tracing. Simple à implanter, elles présentent néanmoins le problème de ne pas diminuer la complexité de la scène. Pour chacun des rayons lancés depuis l'oeil, on calcule une éventuelle intersection avec l'ensemble des englobants de la scène ce qui reste encore très coûteux en temps de calcul. C'est pourquoi ont été développées des techniques d'optimisation diminuant non plus la complexité de calcul mais la complexité géométrique. Ce sont les techniques de décomposition.

1.1.2. Accélération du calcul par diminution du nombre d'intersections rayon-objet.

L'idée principale qui caractérise les méthodes regroupées ici est de diminuer le nombre de rayons étudiés d'un point de vue intersections. Cette approche est indépendante de la complexité mathématique du calcul d'intersection (il est courant dans les intersections de carreaux NURBS de réaliser une approche de la solution par une décomposition du problème). Le premier type d'approche que l'on a présenté à la figure 1 est l'utilisation de volumes englobants hiérarchiques [WHI 80] dont le principe est d'avoir des englobants regroupant des objets géométriquement proches. A l'intérieur de cet englobant, un objet complexe est lui même approché par un englobant général contenant des englobants de plus en plus proches de l'objet. Cette technique permet d'avoir une approximation progressive des objets intersectés par un rayon en généralisant, par une hiérarchisation, la méthode des englobants de Whitted.

On retrouve dans cette classification les méthodes de décomposition de l'espace 3D ainsi que les décompositions de la projection de la scène sur un plan. Les techniques de décomposition de l'espace diminuent considérablement les calculs d'intersections. Elles servent à limiter une recherche d'intersection entre un rayon (primaire, secondaire ou d'ombrage) avec uniquement les objets contenu dans le sous-espace traversé. C'est pourquoi on essaye d'obtenir une décomposition cohérente d'un point de vue géométrique mais aussi "relationnel" entre objets constituant la scène. Glassner [GLAS 84] effectue une décomposition en utilisant les arbres octaux (octree). Il associe à des sous-espaces des propriétés géométriques et des informations de relation d'adjacence permettant de déterminer, pour un rayon, quelles vont être les zones à considérer. De nombreux travaux ont consisté à ne découper la scène qu'en fonction de son contenu, citons pour exemple [HEG 87] [WAR 69] [EXC 87] [BOU 87] [BOU 88] [THI 90] [TOR 90].

Le principe de subdivision de l'espace a été amélioré en introduisant dans la décomposition une analyse sommaire du contenu des sous-régions. Nemoto [NEM 86] adapte les limites de ses subdivisions en faisant "glisser" les frontières des sous-régions sur les axes X,Y,Z afin d'optimiser la répartition à l'intérieur des espaces élémentaires. Les sous-régions étudiées ne sont plus régulières mais le taux d'occupation est plus homogène ce qui globalement diminue les temps de calcul. Des décompositions plus souples (souvent appelées à tort adaptatives) réalisent une subdivision de la scène en analysant la nécessité de décomposer en fonction de critères moins géométriques (complexité, taux d'occupation) [DIP 84] [HEG 87] [JEV 89] [GHA 92] [VIV 91]. On reprendra plus en détail certaines de ces approches dans le paragraphe 3.

Toujours dans l'analyse spatiale, on peut citer les travaux effectués par Devillers [DEV 90] dont l'idée principale peut être résumée en disant qu'elle consiste à rechercher les zones vides d'informations (non présence d'objet) dans lesquelles il sera inutile de lancer des algorithmes coûteux en temps de calcul. Toutes les décompositions spatiales impliquent la création d'algorithmes performants de recherche des zones traversées par les rayons [JEV 89] [KAY 86]. Ces algorithmes sont souvent dépendants de la structure utilisée pour représenter les sous-régions. On décrira cette approche très intéressante au paragraphe 3.

1.2. Optimisation par recherche de cohérence entre les rayons

La seconde catégorie présentée figure 1 concerne les optimisations par diminution du nombre de rayons. L'idée ne consiste pas à diminuer le nombre d'intersections, mais à diminuer le nombre des rayons calculés. Akimoto [AKI 91] propose une méthode basée sur le principe d'interpolation des résultats (méthode présentée au chapitre 4). Il calcule la couleur d'un pixel en réalisant une moyenne pondérée de la valeur des pixels environnants. Le concept de cohérence des rayons est repris par Arvo [ARV 87] et Ohta [OHT 87] mais surtout par Joy [JOY 86] qui pour trouver la valeur d'un point d'intersection entre un rayon et un carreau nurbs utilise une pseudo méthode de Newton en prenant la valeur de l'intersection précédente comme valeur initiale pour faire converger la méthode.

1.3. Généralisation de la notion de rayon

On a regroupé dans cette catégorie un certain nombre de techniques visant à remplacer le concept de rayons fins, utilisé habituellement dans les algorithmes du tracé de rayons, par une notion de faisceau de rayons. Cette idée apparaît conjointement en 84 avec les travaux d'Amatides [AMA 84] qui utilise un faisceau conique et dans ceux de Heckbert [HEC 84], qui utilise un faisceau pyramidal. En 87 Shinya [SHI 87] propose une variante au "Beam-Tracing" le Pencil-Tracing qui permet un calcul plus précis, mais plus complexe, des effets de réfraction. Ces différentes techniques semblent généralement plus rapides qu'un tracé de rayons, car elles permettent de traiter correctement et complètement en même temps les problèmes d'aliassage. On reprendra plus en détail certaines méthodes dans le chapitre 5. On présentera aussi des méthodes d'optimisation et plus particulièrement celle de Ghazanfarpour [GHA 92] qui propose une optimisation de l'algorithme de Heckbert basée sur une décomposition récursive reprenant les concepts développés par Warnock [WAR 69].

Toutes ces méthodes n'ont qu'un seul but véritable: la diminution des temps de calcul qui ont depuis toujours handicapé les algorithmes de rendu réaliste et en particulier celui du Ray-Tracing. On approfondira certaines des techniques décrites succinctement ci-dessus et on montrera comment ces méthodes essayent d'apporter une solution efficace et simple au problème des temps de calcul parfois exorbitants.

2. LES DECOMPOSITIONS

En présentant le principe des volumes englobants on a mis en évidence un problème qui est la recherche d'une intersection entre un rayon et tous les éléments constituant une scène. Les englobants hiérarchiques de Rubin et Witted (WHI 80) apportent une optimisation certaine puisque le raffinement successif des englobants élimine naturellement les objets qui n'ont aucune chance d'être coupés par le rayon. On peut déjà parler d'une diminution de la complexité (en nombre d'éléments à traiter) due à une organisation différente ou une représentation différente du contenu. Cependant, le fait de manipuler des englobants d'englobants entraîne une perte importante de précision et globalement la méthode reste peu efficace. On a alors essayé de résoudre le problème d'une tout autre façon. Plutôt que de créer une "méta-connaissance" de la scène en approchant les objets par des volumes plus importants, pourquoi ne pas essayer d'avoir une représentation de celle-ci à un niveau très petit. Cette restriction du niveau d'analyse simplifie localement la complexité. En d'autres termes, seul un petit nombre d'éléments (voir aucun) occupe une petite partie de l'espace donc pour cet espace le calcul d'intersection avec ces objets sera simplifié.

La différence principale entre les volumes englobants de scène et la décomposition de l'espace est que pour calculer les intersections rayons-objets avec le technique des englobants on réalise *des test sur des objets basés dans un espace connu* alors qu'avec une décomposition on effectue *des tests sur un espace dans lequel on connaît les objets*.

On va présenter maintenant quelques unes des techniques de décompositions spatiales les plus utilisées. Toutes les méthodes que l'on va décrire dans un espace à trois dimensions existent en deux dimensions mais ne sont dans la plupart des cas qu'une restriction de la méthode à un espace de dimension inférieure.

2.1. Les subdivisions non uniformes

Le principe de la subdivision non uniforme est de découper l'espace, ou une représentation de celui-ci, en un ensemble de régions dont la taille varie en fonction des caractéristiques de son environnement. Les caractéristiques prises en compte, pour la subdivision, sont essentiellement géométriques. Cette subdivision consiste à diminuer le nombre d'objets contenus dans les subdivisions. Pour cela, on analyse le contenu d'une fenêtre; si ce contenu ne semble pas être sous une forme simple, on réalise une nouvelle subdivision de la fenêtre et on regarde dans quel sous-espace se trouvent les objets de la scène.

Warnock [WAR 69] a été le premier à proposer une décomposition récursive non uniforme, non pas de l'espace mais du plan. Il projette le contenu de la scène sur un plan pouvant être l'écran. Il découpe récursivement le plan afin de simplifier le contenu des subdivisions. Une des difficultés majeures dans les subdivisions non uniformes est d'estimer le moment où le contenu d'une sous-région devient "simple". Warnock simplifie le problème en distinguant trois types de situations. La décomposition s'arrête quand une sous-fenêtre est soit vide, soit ne contient qu'un seul élément, soit a atteint la taille minimale qui est le pixel, le pixel étant la plus petite partie du plan considérée comme étant indivisible. Le principe de subdivision de Warnock est assimilable à une décomposition de type quadtree sur les composantes X et Y, la dimension en Z restant constante. Cette décomposition avait pour seule ambition d'améliorer le calcul des faces cachées. Pour chaque subdivision, pour chaque rayon lancé depuis l'oeil arrivant dans cette subdivision, on recherche une intersection éventuelle avec l'ensemble des objets contenus dans la sous-fenêtre. Sa dépendance vis-à-vis de la position de l'observateur ne lui permet pas, en l'état, d'être utilisable dans l'implantation un algorithme général de rendu réaliste (rayons primaires, secondaires, d'ombrage).

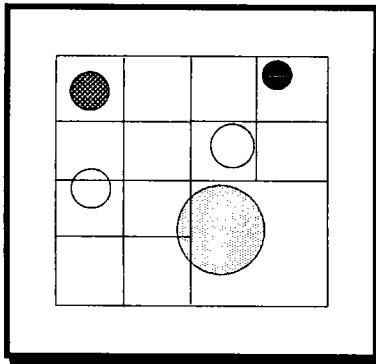


figure 7: décomposition non uniforme

Cette méthode de simplification du contenu a été étendue ensuite à l'espace tout entier. L'arbre des subdivisions n'était plus représenté sous la forme d'un arbre quaternaire mais sous la forme d'un arbre octal (octree) et l'espace est subdivisé en huit sous-régions de taille identique. On conserve les principes de subdivision du plan et l'on arrête la décomposition lorsque chacune des régions est soit vide, soit ne contient qu'un seul objet, soit a atteint la taille minimale : le voxel. Le voxel est la plus petite entité de volume non décomposable dans l'espace observateur. Un voxel n'aura pas la même taille si l'observateur est situé près de la scène ou si au contraire il s'en trouve très éloigné. On ne peut établir de relation avec le pixel, si ce n'est dans la définition.

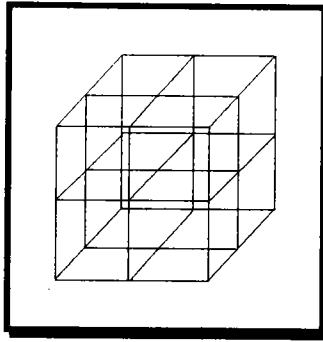


figure 8: une décomposition octale (octree)

Pour chaque rayon l'optimisation du calcul d'intersection se déroule alors en trois parties:

- 1: On lance un rayon depuis l'oeil et l'on recherche la première cellule intersectée par le rayon.
- 2: Si la cellule n'est pas vide, on recherche une intersection avec l'ensemble des objets associés à la cellule.
- 3: Si une intersection existe on associe au point la couleur de l'objet éclairé sinon la couleur associée est par exemple le noir. Si l'objet est spéculaire et/ou transparent, on lance en plus du rayon d'ombrage, depuis le point d'intersection, un rayon réfléchi ou réfracté. On recherche les cellules intersectées par le rayon d'ombrage et les rayons secondaires et le processus recommence.

Les décompositions de type octree présentent deux grands avantages.

Le premier est que contrairement aux décompositions de type quadtree, la subdivision n'est pas dépendante de la position de l'observateur.

Le second, conséquence logique du premier, est que les décompositions spatiales permettent d'optimiser sur le même principe que les rayons primaires, les rayons d'ombrage et les rayons secondaires. Elles présentent cependant un inconvénient majeur qui est de trouver quelles vont être les cellules traversées par le rayon et comment optimiser la propagation de ce rayon (réfracté, réfléchi, ombrage).

De nombreuses approches ont utilisé un découpage non uniforme de l'espace en utilisant une structure à base d'octree [GLAS 84] [KUN 85] ou en mettant en oeuvre des décompositions très différentes comme la création de cellules [WEI 87] [ARN 88] ou la recherche des zones vides [DEV 89]. Citons aussi un ensemble de travaux présentant de décompositions non uniformes [KAP 85] [JAN 86] [PUR 86] [DEV 88] [WYV 68] [PUE 88] [FUJ 85]. Certains ayant combiné les avantages de la partition spatiale et des englobants hiérarchiques.

On présente plus en détail les principales méthodes.

2.1.1. Un exemple de décomposition non uniforme avec un arbre octal: l'approche de Glassner

De nombreuses approches ont été basées sur une décomposition de type arbre octal [SAM 84]. Nous présentons à titre d'exemple significatif celle de Glassner.

Glassner [GLA 84] présente une variation des octree adaptée au tracé de rayons. Il reprend le principe de décomposition de Warnock adapté à une subdivision de l'espace. Il associe à chaque face des voxels une liste des objets dont la surface coupe une des 6 faces de la sous-région. Seuls les objets de cette liste seront considérés pour le calcul d'intersection avec un rayon traversant le voxel (le voxel étant l'extension 3D du pixel, c'est-à-dire la plus petite région indivisible). La liste des objets contenus dans le voxel est en réalité une liste de pointeurs qui ne sont décrits qu'une seule fois en dehors de l'octree. L'octree ne sert qu'à connaître les différentes surfaces présentes dans le voxel. La notion de voxel vide ou voxel plein n'est pas utilisée par Glassner. La scène étant décrite par un arbre de construction, il se peut qu'un voxel pointe sur une surface qui n'existera plus, une fois les opérations booléennes effectuées.

Cette particularité des octree peut sembler un handicap important si l'on désire mettre en oeuvre un algorithme de tracé de rayons pour arbre C.S.G. On montre cependant, avec des mesures peut-être un peu empiriques, que le gain de temps est considérable comparativement à un algorithme de tracé de rayons de base.

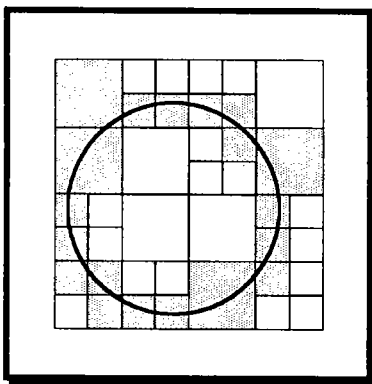


figure 9: décomposition de Glassner

On montre par analogie en 2D (figure 9) une sphère et les voxels de l'octree associés à la décomposition (la décomposition étant ici tout à fait arbitraire). On voit en grisé les voxels coupant les limites de la sphère. L'objet sphère devra être additionné à chaque voxel ombré.

L'algorithme de Glassner présente l'immense avantage de traiter, avec la même décomposition spatiale, aussi bien les rayons primaires que les rayons secondaires et les rayons d'ombrage.

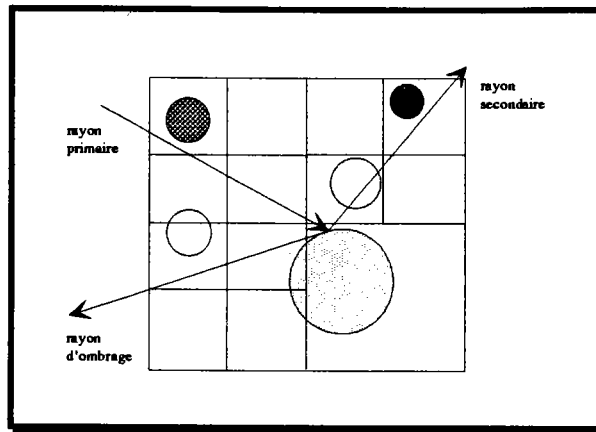


figure 10: parcours des rayons

Comme l'on connaît l'ensemble des objets associés à chacune des sous-régions, il suffit, à partir de la direction des rayons, de trouver les subdivisions traversées et de calculer les intersections avec les objets de la liste. La figure 9 présente le parcours d'un rayon primaire coupant un objet. De cet objet repartent un rayon réfracté et un rayon d'ombrage vers une source lumineuse. On remarque que les mêmes subdivisions sont utilisées pour les trois types de rayons. Il suffit d'utiliser la même procédure de parcours en passant en paramètre les nouvelles valeurs de l'origine du vecteur considéré ainsi que sa direction.

L'algorithme de parcours de la structure est relativement simple dans son énoncé.

procédure Intersection_octree(rayon)

Q = origine du rayon

répéter

localiser le voxel contenant Q {ou coordonnées du voxel traversé par le rayon}

pour chaque objet associé au voxel **faire**

recherche_intersection(rayon, objet)

si pas d'intersection alors

Q = point du prochain voxel intersecté par le rayon

jusqu'à *une intersection soit trouvée ou Q soit en dehors de la structure*

fin

La principale difficulté de mise en oeuvre des décompositions spatiales est d'obtenir un algorithme de parcours des subdivisions performant. Le nombre de sous-régions étant souvent très important, le calcul de parcours représente un handicap. De plus, Glassner ne se contente pas de rechercher la subdivision intersectée, il recherche les faces de la subdivision traversée par le rayon et recherche les intersections effectives avec la liste des objets associés à la face.

2.1.2. Un exemple de décomposition non uniforme avec cellules: La décomposition d'Arnaldi, Priol, Bouatouch

Arnaldi, Priol, Bouatouch [ARN 88] proposent un algorithme de décomposition de l'espace pour des scènes modélisées par un arbre CSG. A la différence des subdivisions non uniformes mais régulières (les sous-régions sont toujours de taille égale) le principe est de diviser l'espace 3D d'une manière non récursive donc non uniforme. La subdivision est réalisée en deux temps, la première étape consiste à déterminer une partition de l'écran. La seconde sert à calculer des englobants minimaux des objets (souvent des volumes parallélépipédiques). On projette ensuite ces volumes sur un plan défini par X et Y (le plan écran). Cette projection génère un ensemble de rectangles. Chaque rectangle est ensuite décomposé suivant ses quatre segments pour réaliser une partition binaire de l'écran sur le principe des BSP tree.

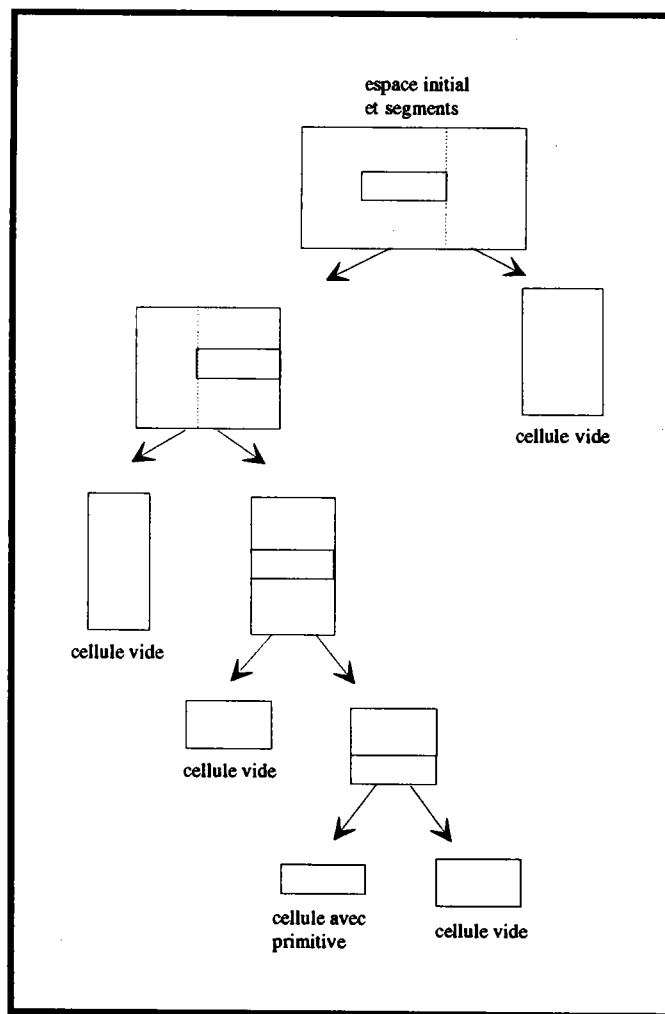


figure 11: séparation binaire de l'espace

Le principe est de réaliser ensuite une extension suivant l'axe des Z

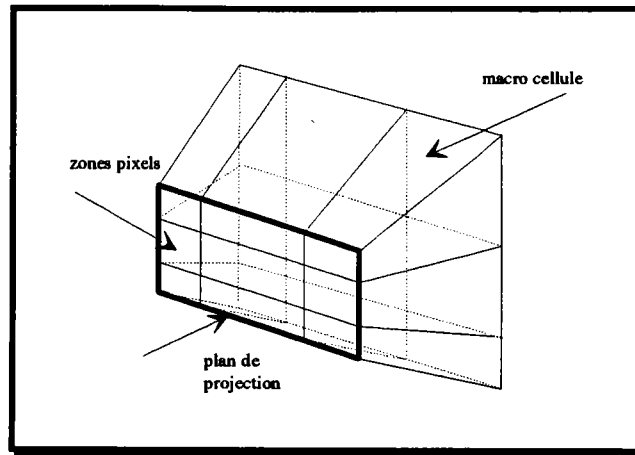


figure 12: création des macro-cellules

On réalise ensuite la même subdivision suivant l'axe des Z, ce qui donne de nouvelles cellules distinctes à partir de la macro-cellule de départ.

Cette approche par une subdivision de l'espace par rapport aux plans de la projection des volumes englobants donne une subdivision irrégulière de la scène. Cette structure de données n'est pas suffisante pour exécuter l'algorithme de lancé de rayons. Il a fallu rajouter à cette structure des informations permettant de localiser la cellule voisine. Pour ce faire, Arnaldi, Priol, Bouatouch ont associé à chaque sommet un ensemble de pointeurs dont le rôle est de relier les différentes cellules par leurs coins.

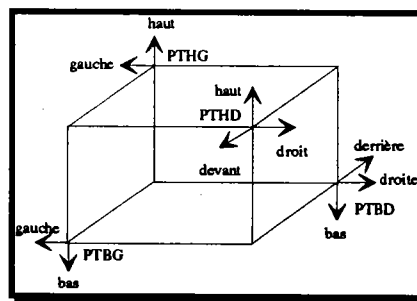


figure 13: connectivité des cellules par pointeur

Les pointeurs de coins permettent de suivre le trajet d'un rayon à l'intérieur de l'espace et de connaître la cellule suivante traversée.

2.1.3. La décomposition non uniforme avec macro-régions: l'approche de Devillers

La complexité des algorithmes de tracé de rayons repose sur deux points, le traitement d'une région et la recherche de la séquence des régions rencontrées par les rayons. Toutes les méthodes présentées jusqu'à présent, considéraient la scène dans son ensemble et cherchaient à en avoir une vue plus simple en subdivisant l'espace. En 89, Devillers [DEV 89][PUE 88] propose une approche fondamentalement différente : les macro-régions. La technique des macro-régions repose sur la construction de sous-espace de dimension quelconque. Dans un premier temps le principe utilise une structure de grille où la taille de la subdivision minimale est choisie de telle manière à ne pas avoir trop d'informations dans les différentes sous-régions. La manière dont sont obtenues les sous-régions est illustrée sur la figure 14. On trouve un cellule vide d'information et l'on essaye d'étendre cette subdivision en X,Y,Z jusqu'à rencontrer des éléments de la grille contenant des objets.

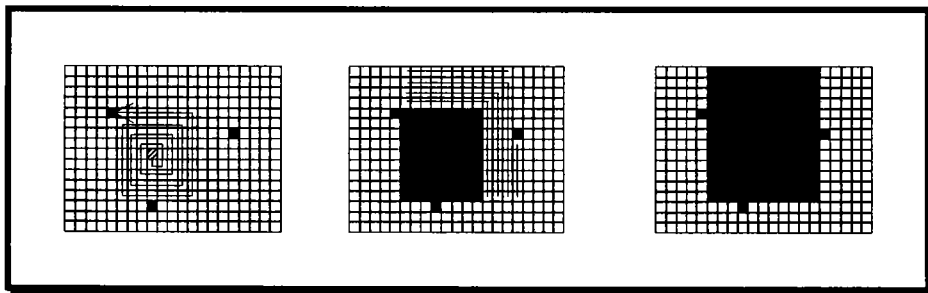


figure 14: recherche des macro-régions

Les macro-régions diminuent le nombre de subdivisions créées pour analyser la scène. Les sous-régions créées ne contiennent pas d'information, elles sont représentatives des zones vides. Le nombre de tests entre un rayon et une sous-région est sensiblement diminué.

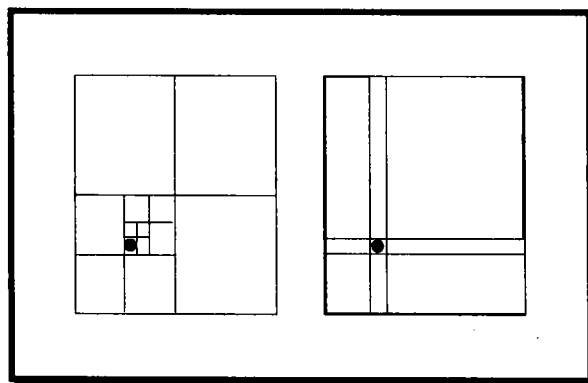


figure 15: comparaison entre les octree et les macro-régions

La traversée d'une structure de type macro-régions est de complexité similaire à la traversée des octree. Elle est cependant plus complexe que le principe de traverser des structures de grille. Il est à remarquer que cet handicap est compensé par un nombre de tests à réaliser beaucoup moins important que dans les cas de subdivisions régulières. Cette traversée est effectuée en deux temps. Si le rayon traverse une des macro-régions, on effectue aucun traitement puisque ces subdivisions sont vides d'informations. Quand le rayon traverse une zone hors macro-région, on utilise la décomposition réalisée avec la grille pour optimiser le calcul d'intersection.

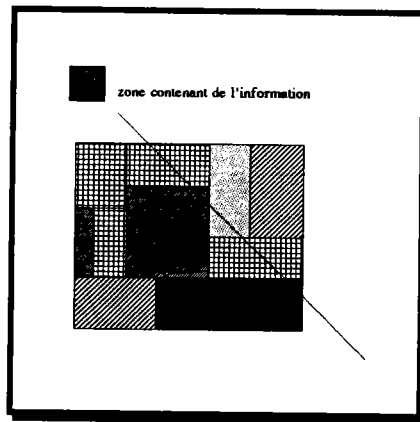


figure 16: parcours d'un rayon avec des macro-régions

On remarque que le rayon traverse trois macro-régions dans lesquelles il ne sera pas utile de rechercher une intersection avec les objets. Quand le rayon quitte une zone vide on repasse à une structure de grill 3D.

2.2. Les subdivisions uniformes

Décomposer une scène de manière uniforme consiste à décomposer l'espace en un ensemble de sous-régions dont la taille est constante (le voxel ou ses multiples). On définit ainsi une grille 3D régulière. Comme pour une subdivision non uniforme, on détermine les objets se trouvant à l'intérieur des subdivisions. On analyse ensuite le trajet du rayon pour savoir quelles vont être les sous-régions traversées. La décomposition de l'espace avec une grille peut se définir par deux caractéristiques importantes: (1) la subdivision uniforme est totalement indépendante du contenu, (2) l'accès au voxel suivant traversé par un rayon s'effectue de manière incrémentale donc très simplifiée. La figure 17 montre les voxels traversés par un rayon. Seul une intersection avec l'objet 1 sera recherchée sur certain voxel.

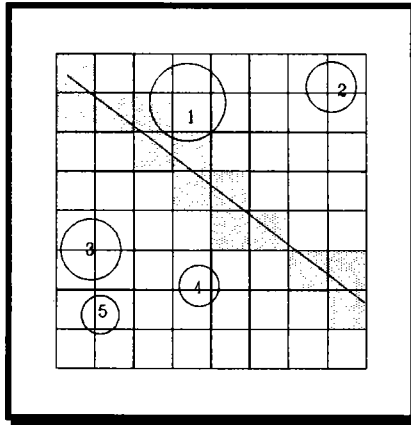


figure 17: décomposition en grille

Le nombre de voxels étudiés est en moyenne plus important que pour une décomposition non uniforme (octree, macro-région, cellules). Cependant l'uniformité des sous-régions a permis de développer plusieurs parcours de sous-régions performants [FUJ 86]. D'autres travaux ont consisté à développer des partitions spatiales avec une grille bi-dimensionnelle structurée en quadtree [GER 86].

2.2.1. Décompositions uniformes: l'approche de Fujimoto, Tanaka et Iwata

Fujimoto et al. [FUJ 86] ont été parmi les premiers à proposer une subdivision spatiale où les voxels, tous de même taille, étaient organisés en grille 3D. L'uniformité des subdivisions a permis à Fujimoto de développer une recherche du voxel suivant basée sur un calcul incrémental. La méthode, appelée 3DDDA (3 dimensional digital difference analyser), reprend le principe de recherche des pixels à allumer pour tracer une droite quand on connaît le pixel de départ et le pixel d'arrivée ("rasterization").

La procédure de recherche est présentée ci-dessous

```

procédure intersection_grille(rayon)
début
  répéter
    calculer i,j,k pour le voxel contenant l'origine du rayon
    mise à jour des valeurs de l'origine_rayon et direction_rayon
    dans la base 3DDDA
  pour chaque objet associé au voxel[i,j,k] faire
    calcul_intersection(rayon,objet)
  si aucune intersection n'a été trouvée alors
    utiliser 3DDDA pour trouver le prochain voxel
  jusqu'à une intersection soit trouvée ou i,j,k en dehors des limites des voxels
fin

```

La figure 18 montre l'ensemble des cellules étudiées pour simuler la trajectoire d'un rayon à l'intérieur d'une structure de grille 3D (représentée ici en 2D). Si l'on se contentait des cellules calculées (représentées sur le schéma par un carré noir) on s'exposerait à des erreurs au niveau du calcul d'intersections puisque des cellules, dans lesquelles passe le rayon, ne sont pas étudiées. Il est donc nécessaire de rechercher et d'étudier l'ensemble de ces cellules (elles apparaissent sur la figure 18 marquées d'un cercle noir). Elles sont déterminées en fonction de la variation sur les axes passifs et actifs.

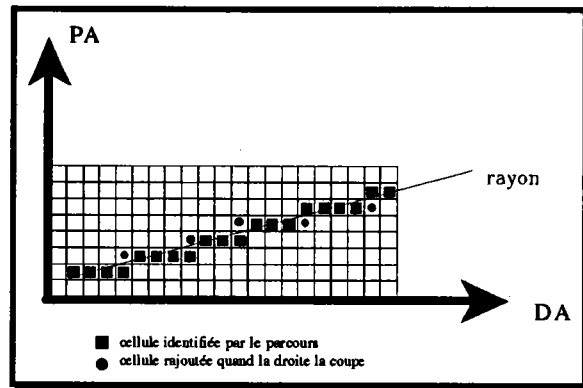


figure 18: incrémentation des cellules

PA représente l'axe passif, c'est-à-dire celui pour lequel il y aura la plus petite variation et DA représente l'axe actif.

On peut citer aussi des travaux très intéressants menés par Cohen and al. [COH 92] sur l'utilisation d'une discrétisation de l'espace.

3. DIMINUTION DU NOMBRE DE RAYONS CALCULÉS

On a présenté dans les chapitres 2 et 3 des méthodes qui visaient à optimiser le calcul de tracé de rayons par une simplification de la représentation de la scène. Des approches tout à fait différentes ont consisté à minimiser le nombre de rayons calculés. La méthode la plus générale consiste à extraire un certain nombre d'informations du calcul d'intersection précédent. L'idée est de déduire la couleur d'un pixel en fonction de la couleur des pixels voisins, comme cela est fait avec des algorithmes d'anti-aliasage, la méthode est simplement étendue à l'ensemble des pixels de la scène [AKI 91]. D'autres travaux ont consisté à diminuer le nombre de rayons calculés par une analyse faite depuis les calculs précédents. Ces dernières méthodes s'appuient essentiellement sur des techniques dites directionnelles comme l'hémicube [COH 84]. Elles sont plus ou moins généralistes, certaines d'entre elles s'appliquent principalement au traitement des rayons d'ombrage [GRE 86].

Nous allons présenter brièvement quelques unes de ces optimisations représentatives des deux grands principes d'optimisation.

3.1. Interpolation: l'approche d'Akimoto

En 91 Akimoto, Mase et Suenaga [AKI 91] proposent une méthode de calcul basée sur une interpolation des valeurs. L'idée est d'étudier des carreaux de pixels de taille constante, d'analyser les coins de ces carreaux et de déduire la couleur des autres pixels en interpolant les valeurs trouvées.

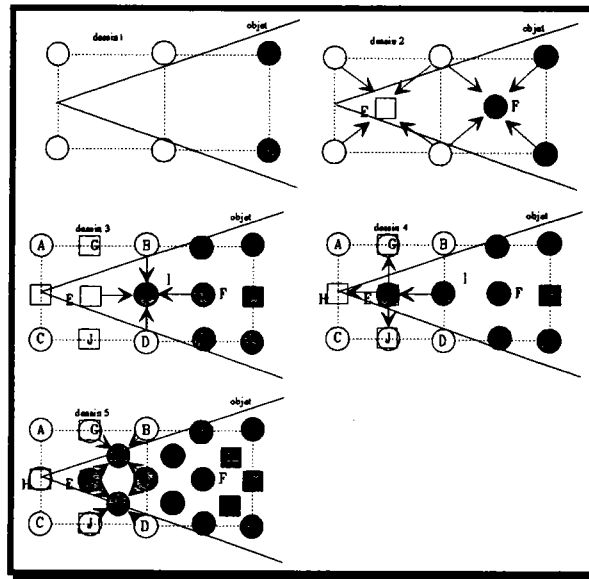


figure 19: interpolation des pixels

La figure 19 donne un descriptif de la méthode. L'objet étant ici le coin d'un prisme

On note par un cercle un pixel calculé, par un carré un pixel déduit, en grisé un pixel à l'intérieur d'un objet, en blanc un pixel vide.

dessin1 : On calcule les pixels des coins. On trouve 4 pixels vides et deux coupant l'objet.

dessin2 : Une première analyse permet de déduire le pixel E puisqu'il est entouré de 4 pixels blancs. Par contre, le pixel F devra être calculé, la couleur des pixels l'entourant étant différente. On remarque que cette analyse, pour l'instant, a engendré une erreur pour E.

dessin3 : On peut à partir du dessin 2 tirer un certain nombre de conclusions sur les pixels de l'environnement de E et F. I étant entouré de pixel grisés et blancs devra, être calculé à son tour alors que les pixels G,J,H sont déduits. On trouve que I appartient à l'objet. Ce qui remet en cause la couleur du pixel E.

dessin4 : On recalcule le pixel E, on trouve qu'il appartient à l'objet, ce qui remet en cause la couleur de J, H et G.

dessin5 : On recalcule ensuite les pixels G,J,H . On calcule ensuite les pixels intermédiaires.

On arrive à une image terminale présentée figure 20.

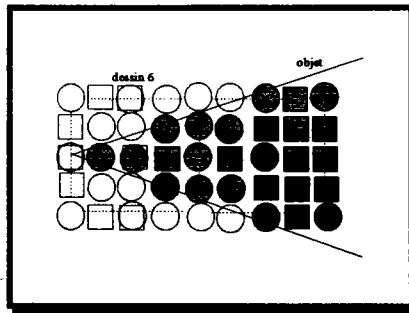


figure 20: interpolation de pixels.

On remarque sur cet exemple relativement simple qu'un grand nombre de pixels peuvent être déduits des pixels périphériques. Le taux d'erreur sur la couleur d'un pixel est relativement faible. Le nombre de pixels recalculés varie suivant la complexité de l'image mais reste inférieur à un pour-cent (936 pixels recalculés pour une image 512 par 512 avec une scène de complexité moyenne).

L'interpolation de pixels est une méthode généraliste qui à l'avantage de traiter l'ensemble des rayons primaires, secondaires et d'ombrage.

Autres applications

Nous citerons aussi pour information les travaux de Joy [JOY 86] qui utilise sur un peu le même principe une cohérence entre les rayons pour optimiser les calculs d'intersections entre un rayon et un carreau NURBS. Il se sert de la valeur de l'intersection précédente comme valeur de départ, il applique une méthode de recherche de solutions d'intersections basée sur le principe de Newton. Comme la nouvelle valeur de l'intersection à trouver risque d'être proche de l'ancienne, la méthode de Newton a plus de chance de ne pas diverger lors des calculs et si cette valeur existe, la méthode convergera rapidement.

3.2. Exemple de technique directionnelle: cube directionnel

Le concept de cube directionnel peut être considéré comme le père de toutes les méthodes d'optimisation basées sur des techniques d'optimisation directionnelles, aussi bien pour les algorithmes de tracé de rayons que pour les algorithmes de radiosit  [COH 84]. C'est une technique qui permet de retrouver les cellules travers es par un rayon sur le m me principe qui consiste, en 2D,   trouver les pixels poss dant une intersection avec une droite. Le cube directionnel est en fait un cube centr  sur l'origine du monde dont les six faces correspondent   six axes dominants ayant comme  tiquette $+X, -X, +Y, -Y, +Z, -Z$.

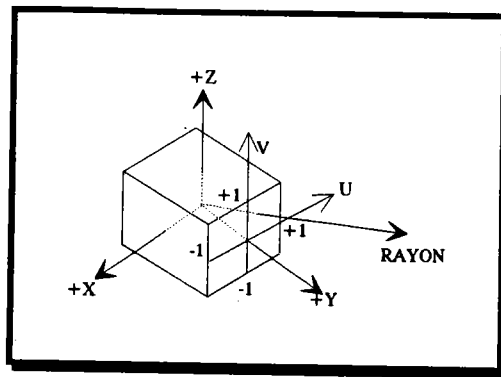


figure 21: le cube directionnel

Le cube directionnel permet de traduire une direction définie en 3D dans un système de coordonnées 2D. On crée pour les six faces du cube un repère U,V. Il devient possible de représenter chaque rayon par les coordonnées 2D en u et v pour chacune des faces qu'il coupe (pour chaque face, les coordonnées seront toujours comprises entre 1 et -1). On projette ensuite l'ensemble des objets sur le cube. Comme l'on connaît les coordonnées du point d'intersection entre un rayon et la ou les différentes faces concernées, on peut facilement trouver la cellule contenant ce point et rechercher une intersection avec les objets associés à cette cellule. Le grand avantage du passage dans un système 2D est qu'il permet d'obtenir soit une décomposition très classique du cube en utilisant une grille soit, c'est ce qui est intéressant, une décomposition adaptative spécifique à chaque face. Cette décomposition permet d'affirmer l'information contenue dans chaque sous-cellule. Il n'existe pas encore d'étude où deux décompositions différentes cohabitent sur les faces d'un même cube, l'écriture d'un algorithme de parcours unique serait difficile à mettre en oeuvre.

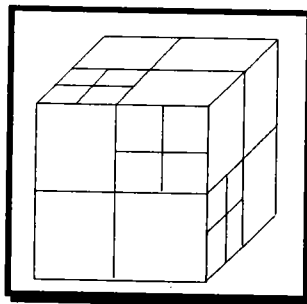


figure 22: décomposition adaptative des différentes faces

Beaucoup de techniques d'optimisation sont basées sur ce concept de cube. Les améliorations des algorithmes ont consisté essentiellement à trouver de nouvelles méthodes de projection de la scène sur le cube ou à trouver des améliorations à la recherche de la cellule à étudier. Le paragraphe suivant présente une des techniques d'optimisation de la recherche.

3.3. Exemple de technique directionnelle: La cohérence des rayons

En 87 Ohta et Maekawa [OHT 87] proposent une méthode utilisant des informations de direction pour accélérer l'algorithme de Ray-Tracing. Leur application est basée sur le théorème de la cohérence de rayons. C'est une technique mathématique qui permet de trouver un volume englobant de la direction d'un rayon qui a une origine dans un objet et son extrémité dans un autre.

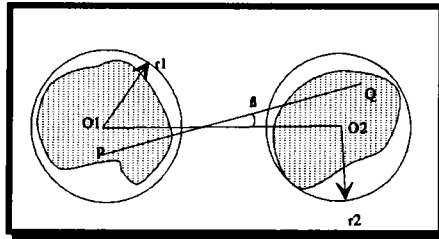


figure 23: Théorème de cohérence

Tous les rayons originaires de la sphère 1 et coupant la sphère 2 définissent un angle β avec la ligne reliant les deux centres des sphères. L'inégalité $\cos \beta > \sqrt{1 - \frac{r1 + r2}{\|O1-O2\|}}$ indique la limite de cet angle par rapport aux valeurs des rayons des sphères et par rapport à la distance entre les centres.

Ohta et Maekawa utilisent ce théorème en l'appliquant non pas à des sphères mais à des englobants convexes polyédriques. Cette extension leur permet d'approcher la scène non plus par des sphères mais par des englobants parallélépipédiques. Sur le même principe que l'hémicube ou le cube directionnel, ils approchent l'espace objets par un englobant formé de cellules de tailles identiques. Un premier calcul rattache à chaque cellule la liste des objets intersectés. L'originalité de leur méthode consiste à utiliser le théorème de cohérence pour optimiser la recherche du parcours d'un rayon.

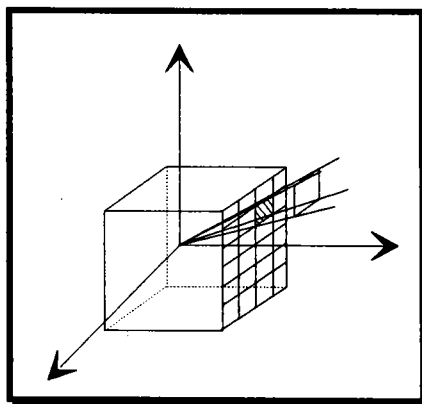


figure 24: subdivision de l'espace

La figure 24 montre le principe simplifié de l'utilisation de la méthode. On calcule un ensemble de distances entre l'origine d'un rayon (oeil, source lumineuse) ici au centre du cube et l'ensemble des objets constituant la scène. On définit, un "volume" de rayons issus de l'origine et coupant chaque objet en utilisant la méthode de cohérence de rayon (*direction volume*). On rajoute ensuite l'objet étudié à chaque cellule du cube intersectée par le volume des rayons.

Autres applications

Nous citerons aussi les travaux de Arvo et Kirk [ARV 87] qui n'utilisent peut-être pas explicitement la notion de cube directionnel si ce n'est dans le traitement des rayons primaires.

4. LA GENERALISATION DE RAYONS

Une des difficultés d'exploitation de la cohérence dans les algorithmes de Ray-Tracing est l'utilisation de rayons infinitésimaux. Cette notion de rayons fins engendre notamment tous les problèmes d'aliasage. Pour remédier à ces deux inconvénients certaines approches ont proposé de généraliser la notion de rayons fins et d'utiliser pour les algorithmes de rendu un faisceau de rayons. Nous présenterons les deux études majeures du domaine et nous citerons certaines optimisations permettant de résoudre, notamment le problème du traitement des objets transparents.

4.1. Généralisation de rayons: l'approche d'Amanatides: Ray-tracing avec cônes

Amanatides généralise la notion de rayon en définissant non plus un rayon fin, mais un cône connu par son sommet, sa direction et son angle d'ouverture. Cet angle est choisi de telle manière que quand un rayon est envoyé depuis l'oeil, il ait, lorsque qu'il arrive sur l'écran virtuel, la largeur d'un pixel. La difficulté de calculer une intersection entre un cône et un objet limite la liste des objets possibles aux sphères et aux polygones. Le calcul d'intersections entre le cône et les objets de la scène se déroule en deux phases.

Pour les sphères la première partie consiste en un rapide test de la position de l'objet par rapport au cône. On recherche le point de la ligne centrale du cône qui est le plus proche du centre de la sphère et calcule la distance entre ces points (SEP). Comme le rayon est un cône on calcule une deuxième distance $D = T \cdot \tan(A) + R / \cos(A)$ ou T est l'origine virtuelle du rayon, A l'angle d'ouverture du rayon et R , le rayon de la sphère. Si la distance D est inférieure à la distance SEP, il n'y a aucune intersection entre le rayon et la sphère. Il est à noter que ce calcul d'intersection comporte deux appels, à des fonctions trigonométriques, ce qui le pénalise par rapport à un calcul d'intersection pour un Ray-tracing conventionnel. La seconde partie de l'analyse, qui n'est pas la moindre, consiste à calculer la quantité de sphère couverte par le rayon. Ceci revient à calculer l'aire de l'intersection entre deux cercles.

Pour le calcul d'intersection avec un polygone on peut appliquer deux stratégies de recherche. La première consiste à rechercher une intersection entre le rayon conique et le plan du polygone puis à trouver l'intersection entre la projection (qui est une conique) avec le polygone. Cette méthode est la plus réaliste mais la plus difficile à mettre en oeuvre. La seconde méthode consiste à projeter les sommets du polygone sur un plan perpendiculaire à la direction du cône. La section projetée du cône sur ce plan est toujours un cercle ce qui simplifie les calculs d'intersections avec la projection du polygone.

Les directions des rayons réfléchis et réfractés sont calculées de la même manière qu'un tracé de rayons conventionnel en utilisant la ligne centrale du cône. Il est nécessaire de recalculer une nouvelle origine ainsi qu'un nouvel angle d'ouverture qui seront dépendant aussi de l'état de surface.

La notion de cône permet de prendre en compte facilement les problèmes de pénombre, de niveau de détail, mais surtout les problèmes d'anti-aliasage qu'il est possible de traiter avec un seul rayon par pixel. De plus la notion de généralisation de rayon à l'avantage de réduire sensiblement le nombre d'intersections calculées.

4.2. Exemple de généralisation de rayons: l'approche d'Heckbert, Hanrahan : le beam tracing

En 84 Heckbert et Hanrahan [HEC 84] introduisent une généralisation de rayons différente: le *beam-tracing*. Ils abandonnent complètement la notion de rayon pour utiliser le concept de faisceau de rayons. Ils appliquent la notion de cône, comme l'a défini Amanatides, au champ visuel tout entier et pour simplifier les calculs, ils ne considèrent non pas un cône mais une pyramide. Un faisceau peut être défini comme un ensemble de rayons ayant tous le même sommet et coupant un certain nombre de polygones plans. La restriction de l'approche est que tous les objets doivent être construits à partir de faces planes ce qui représente une contrainte importante.

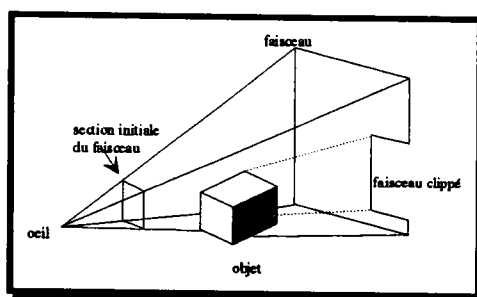


figure 25: le beam-tracing

On recherche l'ensemble des objets se trouvant à l'intérieur de cette pyramide de visualisation. On remarque sur la figure 25 qu'une partie du faisceau est clippée par un objet de la scène. Il est relativement simple d'exprimer cette opération comme la différence de deux polygones dont le résultat reste un polygone. L'analyse réalisée à l'intérieur du faisceau peut être assimilée à celle qui est réalisée dans un Ray-Tracing. Cependant la méthode peut dégénérer et engendrer très rapidement des polygones non convexes ou des polygones contenant des trous.

L'originalité du beam-tracing provient de la façon dont sont traités les rayons secondaires. Le principe est de trouver, sous forme de faisceaux, les zones où vont se propager les rayons lumineux. Plusieurs aspects de l'algorithme du beam-tracing sont identiques à ceux d'un Ray-Tracing standard. On crée un arbre de faisceau de la même manière que l'on crée un arbre de rayons.

La figure 26 illustre le principe pour les rayons réfléchis.

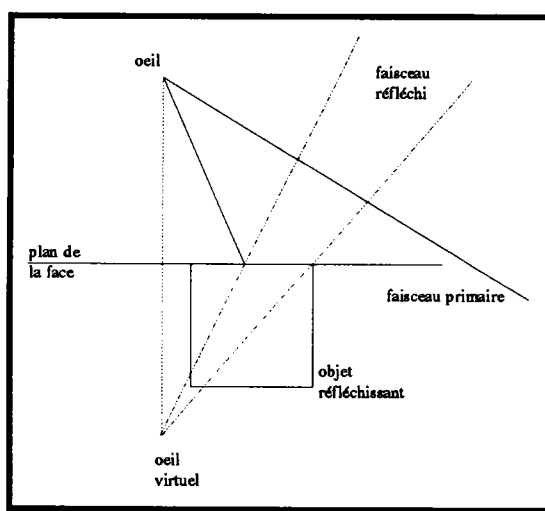


figure 26: un faisceau réfléchi

On trouve pour le faisceau secondaire un observateur virtuel dont la position est symétrique au plan de la face étudiée, l'angle d'ouverture du nouveau faisceau est donné par le clipping réalisé sur cette même face.

La méthode, bien que très agréable, présente cependant une faiblesse importante dans le traitement des rayons réfractés. La réfraction ne conserve pas la nature du faisceau. Dans le cas d'objets non minces la non linéarité du milieu traversé provoque une déformation du faisceau, la notion de pyramide n'existe plus et il devient impossible de créer un observateur virtuel et la méthode s'effondre. Pour remédier à cela, Heckbert et Hanrahan proposent deux solutions. La première est de supposer l'observateur à l'infini et de considérer que les rayons qui arrivent sur la face sont parallèles entre eux. A la sortie de la face les rayons sont "déplacés mais plus déformés". La deuxième solution est de considérer qu'il n'y a aucune déformation quand les rayons traversent un objet transparent.

Ces deux solutions, basées sur des hypothèses fortes, sont très restrictives et sont faites au détriment du réalisme. Ghazanfarpour [GHA 92] apporte une solution partielle au problème des rayons réfractés pour le beam-tracing. Il réalise une décomposition récursive du faisceau jusqu'à considérer le sous-espace obtenu comme linéaire d'un point de vue optique (parfois jusqu'au 16ème de pixel). La décomposition abusive entraîne un surcoût important de temps de traitement et d'utilisation de place mémoire. De plus les contraintes fortes qu'il impose à la modélisation des objets (uniquement des faces planes) font que sa solution ne peut être utilisée que pour un nombre de scènes trop restreint.

Autre approche

En 87 Shinya, Takahashi et Naito [SHI 87] proposent une approche qui s'appuie à la fois sur le Ray-tracing, sur l'interpolation de rayons et sur le beam-tracing : le pencil-tracing. La méthode est basée sur la théorie de l'approximation "Paraxiale" qui permet de considérer la réflexion et surtout la réfraction, dans un milieu homogène, comme linéaires. La mise en oeuvre des outils mathématiques semble cependant assez délicate et pas toujours très facile à assimiler.

5. OPTIMISATION POUR ARBRE CSG

L'arbre CSG permet de construire des solides par combinaisons d'objets primitifs. Les opérations d'assemblages reposent sur les trois opérateurs booléens que sont : *l'intersection (&)* *l'union (+)* et *la différence (-)*. Les optimisations des algorithmes de Ray-Tracing appliqués aux arbres CSG s'articulent autour de deux grands axes: les volumes englobants et la cohérence dans la construction.

L'utilisation des volumes englobants pour les arbres CSG est tout à fait comparable à la méthodes des englobants décrites au chapitre 2. Elle a été introduite par Roth en 82 [ROTH 82] et se limitait à une boîte 2D ou une sphère 3D. La notion de boîte 2D n'était utilisée que pour l'optimisation des rayons primaires. La sphère pouvait être utilisée pour tous les types de rayons . En 84 Glassner [GLA 84] attache à chaque feuille la liste des objets qui coupe le voxel (voir pour plus de détail le paragraphe 3.1.1)

Il est possible de diminuer le nombre de rayons calculés en utilisant les caractéristiques spécifiques des opérateurs de construction d'un arbre CSG [GER 86]. Si l'on recherche une intersection entre un rayon et la différence (ou l'intersection) de deux objets A et B, si le rayon ne coupe pas l'un des deux objets il ne pourra couper la différence (ou l'intersection). Ceci revient à réutiliser les propriétés des englobants en les généralisant avec des opérations booléennes.

Enfin, certains travaux ont consisté à mélanger aux arbres C.S.G des structures permettant de réaliser simplement des optimisations par simplification locale de la complexité de la scène. On présente maintenant une méthode utilisant à la fois une structure octree et une construction de type C.S.G

5.1. Utilisation des arbres C.S.G : L'approche de Wyvill et Kunii

En 85 Kunii et Wyvill [KUN 85] proposent d'utiliser l'octree pour en plus effectuer les opérations booléennes qui jusqu'à présent était effectuée par le tracé de rayons. Il décrivent initialement une scène à l'aide d'un arbre C.S.G. Dans leur version des octree ils définissent un voxel comme pouvant être soit plein, soit vide soit de type contour ou encore de type rebut.

Un voxel de type "contour" est intersecté par l'enveloppe d'un seul objet, cet objet doit être ajouté à la liste des objets à étudier. Si le voxel est dit plein, l'objet contenu n'a pas besoin d'être mémorisé. En effet la notion de plein indique que le voxel se trouve à l'intérieur de la matière et un rayon coupant cet objet aura déjà intersecté son contour avant d'atteindre ce voxel.

Enfin ils définissent un voxel qui ne sera ni plein, ni vide, ni de type contour : le voxel rebut. Ce voxel aura atteint un taille trop petite pour être à nouveau subdivisé et son contenu n'est pas unique.

Cette représentation permet de définir un seul voxel pour représenter un objet non composite. L'octree d'un scène sera une suite d'opérations booléennes sur les octree des objets non composites. Toutes les opérations booléennes seront exprimées en fonction du complémentaire et de l'intersection.

Si la complémentarité ne présente pas de difficulté majeure, l'intersection semble plus délicate à manipuler.

Une intersection entre deux noeuds s'effectuera recursivement sur leurs fils jusqu'à arriver peut-être à une feuille. Une intersection entre un voxel vide et un autre voxel sera vide. Une intersection entre un voxel plein et un autre voxel donnera l'autre voxel. Une intersection entre voxel rebut et un autre voxel (vide ou plein) donnera le voxel rebut. L'intersection entre un voxel rebut et un noeud ne pourra pas se produire. tous les autres cas se régleront par division récursive du voxel.

En 86 Wyvill et al. [WYV 86] proposent une décomposition de l'espace pour accélérer les algorithmes d'intersection. Les tests d'arrêt sont plus complexes qu'une subdivision spatiale avec un autre modèle. La difficulté est de conserver, pour les subdivisions, les contraintes de construction de l'objet global.

Autres approches

En 84 Bronswoort et al. [BRO 84] définissent une orientation différente pour l'optimisation des algorithmes de Ray-Tracing appliqués au CSG. Ils proposent une alternative proche des algorithmes de scan-line.

En 89 Samet [SAM 89] propose d'utiliser une décomposition spatiale (de type octree) comme base d'optimisation et développe un algorithme de subdivisions successives dépendant de la trajectoire du rayon.

6. COMPARATIF DES METHODES

On propose un comparatif des différentes méthodes que l'on a présentées. Cette comparaison est à prendre avec beaucoup de réserves. Il est en effet souvent très difficile d'obtenir de la part des auteurs une grande impartialité dans l'interprétation de leurs résultats et il est bien rare de voir apparaître dans les différents articles les problèmes pour lesquels l'auteur n'apporte pas de réponse ou une réponse incomplète. C'est pourquoi nous présentons une comparaison entre les grandes familles d'optimisation. Il est à noter aussi que ce tableau n'est valable que pour l'algorithme de tracé de rayons, certaines méthodes comme le cube directionnel appliqué à la radiosité (hémicube) ne servent plus à traiter un ensemble de rayons puisque cette notion n'existe pas pour les algorithmes basés sur des échanges entre surfaces.

	englobants	subdivisions non uniformes	subdivisions uniformes	cube directionnel	interpolation de pixels	rayons généralisés
mise en oeuvre	simple	moyenne	simple	complexe	moyenne	complexe
application aux modèles	tous	tous (ramené au plan pour les NURBS)	tous (ramené au plan pour les NURBS)	polygonal	tous	tous (seul polygonal pour le beam-tracing)
rayons traités	tous	tous	tous	tous	tous	tous (difficile pour les réfractés)
mémoire utilisée	faible	importante	importante	importante	moyenne	importante
performance	moyenne	bonne	bonne	bonne	bonne	bonne

Mise à part une optimisation basée essentiellement sur les englobants qui possède des performances sensiblement inférieures, il est bien difficile de comparer d'un point de vue performance pure toutes les autres méthodes. Deux explications à cela, la première est que l'on ne connaît que trop rarement les algorithmes de bases et les structures de données réellement utilisées, la seconde est que l'on ne connaît pas la manière dont a été modélisée la scène test. Ce qui rend la comparaison des méthodes trop approximative et trop dépendante de la programmation. C'est pourquoi on ne présente, dans le tableau, que des tendances. Il aurait fallu, pour avoir une critique plus profonde et plus constructive, programmer toutes les méthodes d'optimisation et les comparer avec la même structure de données ce qui n'était pas le but de cette étude.

7. LES OPTIMISATIONS FUTURES

Les optimisations futures vont devoir intégrer, dans l'analyse qui sera faite de la scène, un certain nombre de caractéristiques propres à la façon dont seront modélisés les objets, leur influence d'un point de vue géométrique mais aussi lumineux, fonctionnel, physique. La construction des volumes englobants (hiérarchiques ou non), les décomposition devront s'adapter en fonction de la nature des éléments constituant la scène. Les optimisations devront s'appuyer sur un ensemble d'informations extraites automatiquement de la scène étudiée. L'acquisition et la réutilisation de connaissances sera le point de passage obligé pour améliorer encore les outils et les temps de calcul des algorithmes de rendu réaliste. Il ne devrait pas être nécessaire de recalculer toute une image si la seule modification se situe au niveau d'un objet peu influant d'un point de vue géométrique ou lumineux.

Bien que peu présentée ici la parallélisation d'algorithmes devrait ouvrir, pour des domaines très ciblés, des perspectives intéressantes. La parallélisation automatique d'algorithme est une chose encore mal maîtrisée, même en image de synthèse, elle ne se limite pas à multiplier les microprocesseurs de traitements afin de diminuer les calculs.

Restent enfin toutes les optimisations mathématiques sur le calcul d'intersections, la modélisation, la perception même de la notion de rayon ainsi que toutes les améliorations du matériel qui intègre de plus en plus de fonctions graphiques câblées.

Chapitre 3

DEFINITION D'UNE APPROCHE ADAPTATIVE

1. INTRODUCTION

L'analyse des travaux existants pour l'amélioration des performances de l'algorithme de tracé de rayons et du tracé de faisceaux montre un certain nombre d'avantages, mais également de faiblesses pour chaque méthode, parmi lesquelles :

- Les résultats de la mise en oeuvre de méthodes cherchant à diminuer le nombre d'intersections rayon-objet, dans l'algorithme de tracé de rayons, peuvent être plus ou moins intéressants selon les types d'objets constituant la scène et surtout leur répartition dans l'espace.

- Les méthodes basées sur le "beam tracing" impliquent des scènes à facettes planes, des approximations parfois peu justifiables théoriquement sur les rayons secondaires et sur les traitements qui deviennent complexes.

Pour tenter de pallier ces défauts, nous avons cherché à développer une approche différente, dans laquelle les termes les plus significatifs sont : décomposition, propagation de faisceau, adaptabilité et interactivité [GAR 90][GAR 91][GAR 93]. En effet, nous considérons qu'une décomposition ne donne des résultats intéressants que si elle peut s'adapter (soit automatiquement, soit par une aide de l'opérateur) à la scène traitée, que ce soit au sens type d'objet (faces planes, faces gauches ...) qu'au sens répartition dans l'espace.

Avant de détailler les différents aspects de la méthode, nous en donnons ci-dessous les grandes lignes.

La méthode de décomposition adaptative repose sur un concept de multi-analyse.

Nous effectuons un prétraitement basé sur le principe du tracé de faisceaux. En propageant un cône de visualisation, nous délimitons des zones de l'espace à l'intérieur desquelles nous pouvons concentrer l'étude. Initialement, ce cône représente le champ de visualisation de l'observateur principal; il est ensuite étendu à l'étude des réflexions ou des réfractions suivant les caractéristiques optiques des objets rencontrés. Utilisant des techniques de propagations de faisceaux similaires à celles mise en oeuvre dans

l'algorithme du beam-tracing, nous connaissons les mêmes limites liées à la forme des objets. Afin de ne pas se limiter à des scènes modélisées uniquement par des faces planes, nous choisissons de modifier les optimisations en fonction des caractéristiques géométriques des objets étudiés. Pour cela, les techniques d'optimisation doivent s'adapter automatiquement à l'objet analysé.

Si celui-ci est constitué de faces planes, le principe d'optimisation retenu sera une décomposition que nous appelons adaptative et que nous présenterons plus en détail par la suite, sinon nous optons pour une forme d'optimisation plus conventionnelle comme par exemple les englobants. Nous sommes conscient que les englobants ne représentent pas la meilleure optimisation possible des rayons secondaires et d'autres techniques, comme par exemple une utilisation d'octree, seraient plus adaptées.

Nous avons préféré valider le principe de choix automatique entre différents algorithmes même si les options retenues pour certain traitement ne semblaient pas toujours les plus adaptés.

Pour réaliser l'affichage des objets de la scène, nous appliquons, lors d'une seconde étape, à l'intérieur des cônes de visualisation un algorithme de tracé de rayons. Afin de ne pas retrouver les médiocres performances du tracé de rayons, nous effectuons, à l'intérieur des cônes de visualisation une décomposition récursive diminuant localement la complexité des scènes analysées. Cette décomposition de type quadtree est appliquée non pas à la projection des objets mais à la projection des englobants sur un plan de référence qui peut être l'écran pour l'observateur principal ou le plan d'une face pour le traitement des rayons secondaires. Le fait de projeter les englobants permet de reconstituer simplement des rectangles dont les cotés seront parallèles aux bords de l'écran (ou de la face). Ces sous-régions seront ensuite assimilées à des "sous-écrans" à l'intérieur desquels nous limiterons l'application du tracé de rayons. La création de ces différentes sous-espaces va permettre une application sélective du tracé de rayons. Le principe de subdivision de la fenêtre de travail reste conventionnel, il partage la zone d'étude en quatre sous-parties de taille identique. Cependant, les tests d'arrêt mis en oeuvre prennent en compte un ensemble de paramètres liés à la position des objets, à leur taille ainsi qu'à leur concentration. Nous avons voulu définir une méthode d'analyse qui puisse adapter la décision de poursuivre ou non une subdivision en réalisant une synthèse d'un ensemble d'informations provenant d'une suite d'analyses réalisées indépendamment les unes des autres. La multiplicité des informations prises en compte ainsi que la combinaison des tests d'arrêt va permettre d'avoir un certain recul vis-à-vis de la scène analysée et d'interpréter différemment les conclusions des analyses.

Pour affiner l'étude par rapport à celles réalisées habituellement dans une décomposition récursive de type quadtree, nous avons défini deux grandes familles de tests dont les fonctionnalités seront à la fois différentes et complémentaires.

La première catégorie regroupe des tests dits terminaux, leur rôle va être de suspendre la décomposition. Contrairement aux tests mis en place par Warnock dans sa décomposition, ces tests ne vont pas servir à guider la subdivision. Ils servent à détecter des situations particulières mais ne sont aucunement représentatifs de situations recherchées. Les tests terminaux, au nombre de trois (cf.

paragraphe 6), vont servir à éviter des décompositions abusives pour des situations où la complexité de la scène (nombre d'objets, concentration, ...) ne pourra être ramenée à un niveau acceptable par des subdivisions supplémentaires. Nous aurons l'occasion de montrer au chapitre 4 que la décomposition importante aura tendance à pénaliser, parfois lourdement, les temps de calcul.

La seconde catégorie regroupe des tests dits non terminaux; ils vont orienter l'algorithme parfois vers l'arrêt de la décomposition, parfois vers la poursuite. Pris séparément, ils ne peuvent influencer directement sur la décision terminale. C'est en fait la combinaison des valeurs de ces différents tests qui va décider ou non de l'arrêt. Pour chacun de ces tests nous avons défini deux paramètres.

Le premier représente la prise en compte du test. Par la modification d'un booléen on peut, lors de la décomposition, ignorer ou non l'un des tests. Il existe des scènes où il est inutile de considérer l'ensemble des tests d'arrêt ou d'orientation.

Le second paramètre représente la valeur associée au test (nombre d'objets dans une sous-fenêtre ou encore la taille de la fenêtre minimale).

Le principe d'application de l'algorithme de tracé de rayons que nous utilisons pour la décomposition adaptative et que nous expliciterons plus en détails au paragraphe 4, permet de traiter, sans diminution des performances, des sous-fenêtres contenant un nombre d'objets plus important que dans des décompositions classiques.

Le premier test non terminal que nous avons défini est donc le nombre d'objets maximal à l'intérieur d'une sous-fenêtre. Pour l'ensemble des images que nous avons calculées le nombre d'objets par sous-fenêtres n'était jamais inférieur à 5, en dessous de ce seuil les performances de calcul avaient tendance à se dégrader ou tout du moins à ne pas être meilleurs (voir chapitre 4: Résultats et bilan).

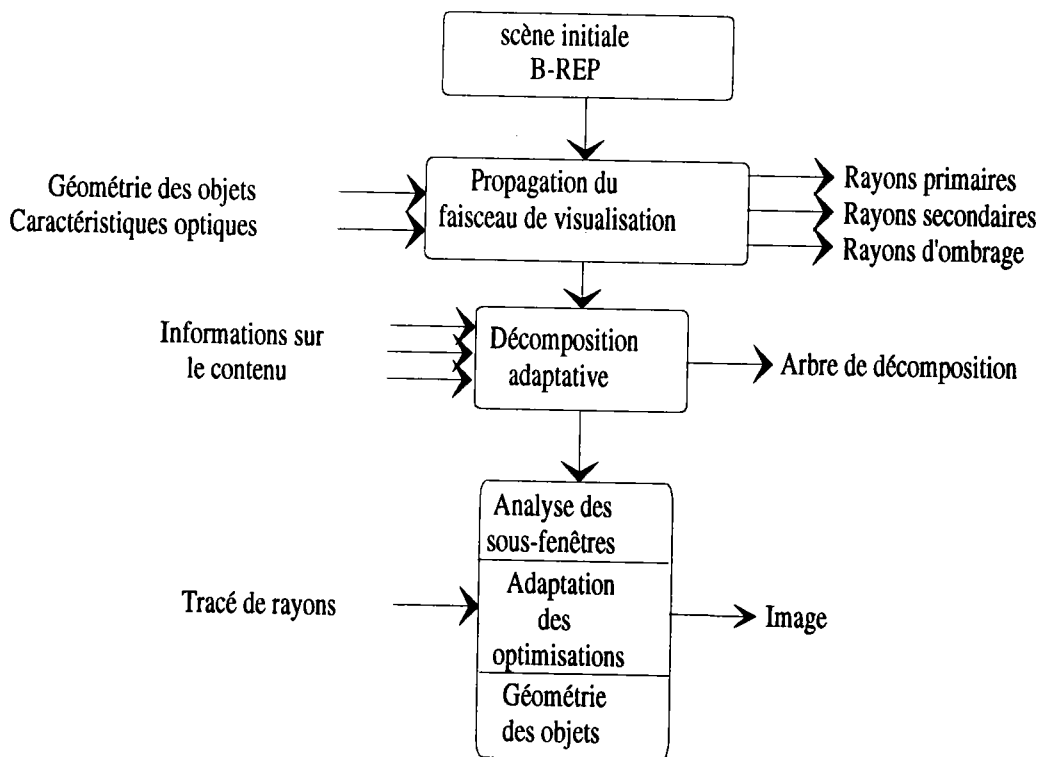
Le second test utilisé est le taux d'occupation de la fenêtre. On essaye, par une analyse de la surface occupée par la projection des objets, de prévoir les possibilités d'intersection entre les objets. Si, par exemple, le taux d'occupation est faible par rapport à la taille de la fenêtre et si le nombre d'objets est proportionnellement important on peut imaginer que beaucoup d'entre eux risquent de se couper. Poursuivre la subdivision pour isoler des cas plus simple peut s'avérer inutile, la décomposition peut très vite atteindre des niveaux importants sans pouvoir mettre en évidence des zones où l'application d'un algorithme de tracé de rayons s'avèrera plus efficace. Il est évident que nous n'effectuons qu'une interprétation de valeurs, les conclusions peuvent être à tout moment mises en défaut par une situation particulière.

Néanmoins nous essayons, à partir de toutes les informations recueillies lors de l'analyse de la scène et des connaissances liées aux modèles, d'effectuer une synthèse et de prendre une décision en fonction du meilleur compromis possible entre les temps liés à la décomposition et l'optimisation du calcul des images. Nous avons constaté que dans la grande majorité des cas, les interprétations des valeurs de tests permettent d'obtenir une représentation fidèle de la situation.

Nous avons souhaité exploiter les connaissances, pas obligatoirement très pointues, de l'opérateur. Nous lui avons laissé la possibilité de modifier les différentes valeurs influant sur la décomposition.

Il est évident qu'une déduction automatique des meilleurs valeurs de paramètres en fonction de la situation analysée serait le plus souhaitable. Cette éventualité fait partie des améliorations futures de la méthode encore faudra-il que les temps consacrés à l'examen de la scène ne soit pas plus importants que les temps d'affichage. Pour l'instant c'est l'opérateur qui joue le rôle de système expert (ayons encore confiance dans les valeurs de l'Homme). Il a la possibilité, avant chaque calcul d'image, de modifier l'ensemble des paramètres des tests. Il peut diminuer ou augmenter les valeurs du nombre d'objets par sous-fenêtre et du taux d'occupation en fonction du nombre d'objets de la scène et de leur concentration.

Le schéma synoptique ci-dessous présente les différentes étapes d'une analyse adaptative



Après avoir exposé brièvement les limites des décompositions classiques, nous présenterons plus en détails les différents modules du schéma de décomposition. Nous définirons une adaptabilité à la forme et à la position de la projection des objets pour les rayons primaires, secondaires et d'ombrage. Nous proposerons ensuite une définition plus complète des tests d'arrêts terminaux et non terminaux. Nous expliquerons comment ils se combinent en fonction des différentes situations et quelles sont les informations importantes que nous pouvons en extraire. Nous proposerons enfin une solution pour adapter les méthodes d'optimisation aux caractéristiques géométriques et optiques des objets.

2. LIMITES DES DECOMPOSITIONS CLASSIQUES

Dans son article, Whitted [WHI 69] présente une méthode de tracé de rayons dont la seule optimisation, ô combien importante, est l'utilisation d'englobants. L'algorithme, baptisé "algorithme de la force brutale", est incapable de tenir compte d'une quelconque répartition des objets dans l'espace. Il n'existe pas, dans la version d'origine, une analyse du contenu qui permette de prendre en compte des caractéristiques de positionnement afin d'optimiser les calculs. Un rayon est calculé pour chaque pixel et une intersection est recherchée pour tous les objets de la scène.

L'algorithme s'écrit:

Algorithme 1

```
    Pour tous les pixels de l'écran faire
        Pour tous les objets faire
            Calculer l'intersection rayon-objet
        Fin pour
    Afficher l'objet le plus proche
Fin pour
```

Pour remédier à ce problème, comme nous l'avons présenté au chapitre 2, ont été développées de nombreuses optimisations consistant à effectuer une décomposition spatiale ou surfacique de l'espace de travail. Les temps de traitement ont fortement diminué mais ces optimisations sont encore très loin de prendre en compte des critères de position d'objets dans l'espace. En effet, on s'aperçoit que le problème de balayage de tout l'espace, lié à un manque d'information, que nous avons pour l'algorithme de la force brutale, se retrouve au niveau de l'analyse des subdivisions. Les décompositions récursives ne servent aucunement à préciser la localisation des objets, elles contribuent uniquement à diminuer localement la complexité de la scène. Illustrons ces propos en analysant l'exemple simple d'une scène contenant deux objets sur laquelle on applique une décomposition récursive de type Warnock (figure 1). Une telle décomposition permet de mettre en évidence des zones simples à traiter, à savoir les zones vides et celles ne contenant qu'un seul objet. Par contre un nombre de sous-fenêtres contiendra toujours deux objets même si la subdivision est poursuivie.

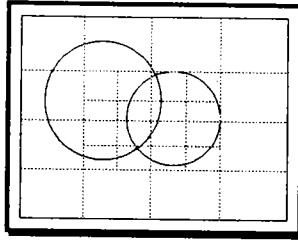


figure 1: scène de deux objets

Pour cet exemple l'algorithme de tracé de rayons s'écrit :

Algorithme 2

```

    Pour chaque sous-fenêtre non vide faire
      Pour tous les pixels de la sous-fenêtre faire
        Pour tous les objets de la sous-fenêtre faire
          Calculer l'intersection rayon-objet
        Fin pour
      Afficher l'objet le plus proche
    Fin pour
  Fin pour
  
```

L'application de cet algorithme revient à considérer chacune des sous-fenêtres comme un écran sur lequel on applique l'algorithme de la force brutale en évitant de traiter les subdivisions vides. Ce qui ne représente finalement qu'une très petite amélioration de l'algorithme de Whitted, d'un point de vue compréhension du contenu de la scène. Cette remarque reste valable pour la décomposition de type spatiale (octree, macro-régions).

En effet, le principe est de lancer un rayon et de voir si le sous-espace traversé par ce rayon contient ou non des objets. On traite encore tout un ensemble de rayons qui ne va apporter aucune information importante à la construction de l'image. Il existe des cas trop simples avec lesquels on peut très facilement mettre en évidence les carences de tels approches (figure 2).

On s'aperçoit qu'une décomposition classique est ici complètement inadaptée à la constitution de la scène et à la répartition des objets dans l'espace de travail. En effet, la superposition partielle ou totale de deux objets provoque une décomposition abusive d'une zone qu'il sera impossible de simplifier. L'algorithme de tracé de rayons devra être appliqué à un ensemble de sous-fenêtres de contenu identique. L'apport de la décomposition, sur l'optimisation des calculs, aura été de très faible importance. De plus les coûts de temps et de place mémoire engendrés par une décomposition excessive influent négativement sur les temps de calcul.

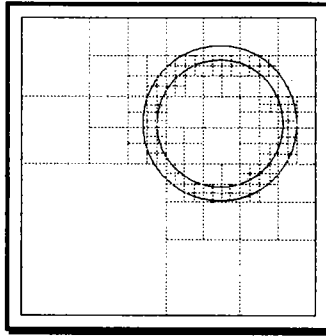


figure 2: décomposition adaptative

La plupart des méthodes de décomposition ne permettent pas d'obtenir, seules, tout un ensemble de caractéristiques liées à la position des objets. Ces informations permettraient une analyse plus en profondeur du contenu et une application de l'algorithme de tracé de rayons uniquement aux zones contenant de l'information.

C'est pourquoi nous avons choisi de développer, en plus des décompositions, une méthode simple qui apporte un ensemble de renseignements sur les relations entre objets et leur localisation dans l'espace.

3. DEFINITION D'UNE APPROCHE ADAPTATIVE

Nous avons souhaité définir une nouvelle approche où l'analyse de contenu de la scène soit indépendante de l'application de l'algorithme de rendu. La méthode se déroule en deux phases bien distinctes où les termes importants seront: analyse, adaptabilité, interactivité.

La première partie de la méthode consiste en une étude de la répartition, de la forme et des caractéristiques optiques des objets. Nous propageons une pyramide de visualisation délimitant ainsi des zones à l'intérieur desquelles nous limiterons l'application de l'algorithme de rendu.

La seconde partie consiste en l'application de l'algorithme de tracé de rayons en prenant en compte des connaissances extraites de l'analyse du contenu. Afin d'augmenter les performances de calcul de l'algorithme de tracé de rayons, nous effectuons à l'intérieur des pyramides de visualisation une décomposition récursive. L'adaptabilité se retrouvera dans le choix des optimisations mises en oeuvre pour l'algorithme de tracé de rayons. Nous adapterons automatiquement ces optimisations aux caractéristiques des objets étudiés.

、 Pour cela, la méthode devra tenir compte d'un ensemble de critères qui vont orienter constamment les décisions pour la poursuite de la décomposition et pour le choix des optimisations.

4. ADAPTABILITE A LA POSITION ET A LA FORME DES OBJETS: APPLICATION AUX RAYONS PRIMAIRES

L'ensemble des scènes sur lesquelles nous avons travaillé, ont été réalisées à partir d'un modèleur B-REP. Tous les objets sont connus par un englobant sphérique sur les objets et/ou sur les faces¹. Nous avons cherché un moyen permettant à l'observateur de déterminer qu'elles vont être les zones importantes du point de vue du concentration et présence d'objets. Pour cela nous avons réalisé une projection perspective de chaque sphère englobante sur un plan (représentant l'écran pour les rayons primaires).

Le problème revient à calculer l'intersection entre un plan et le cône de projection de la sphère ayant comme sommet l'oeil. Afin de simplifier ce calcul nous nous contentons des quatre points de tangence entre la sphère et une pyramide ayant toujours l'oeil comme sommet (figure 3). Ces quatre points délimitent plus simplement une zone représentative de la projection de la sphère.

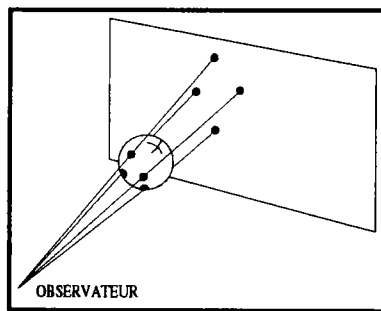


figure 3: projection de points tangents

On englobe ensuite la surface projetée par un rectangle dont les côtés sont parallèles aux bords du parallélépipède rectangle (représentant l'écran ou une face étudiée) et passant par ces quatre points projetés (figure 4). La surface ainsi définie représente une approximation par excès du volume englobant de l'objet. Une définition des englobants par des boîtes aurait sans aucun doute été plus performante, elle aurait permis de mieux approcher la forme de l'objet et de diminuer sensiblement la fenêtre de projection.

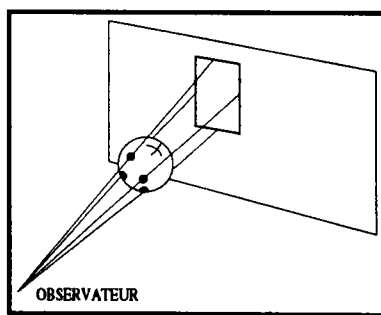


figure 4: rectangle de projection

¹ contraintes imposées initialement par T.D.I

Nous sommes en mesure de limiter l'application de l'algorithme de tracé de rayons en tirant parti de la projection des englobants. En effet, nous pouvons maintenant définir une zone restreinte dans laquelle nous pourrions limiter le calcul des rayons primaires et indirectement des rayons secondaires. La projection est assimilée à une sous-écran, tous les rayons qui, initialement étaient lancés inutilement en dehors de cette zone, ne seront plus calculés. Ceci est sensiblement différent du traitement effectué habituellement avec les décompositions. En effet, la localisation d'information par une restriction de l'espace d'application inverse, pour les rayons primaires, l'ordre des opérations de l'algorithme de tracé de rayons. Une analyse conventionnelle regarde pour chaque pixels d'une sous-fenêtre de décomposition quels sont les objets susceptibles d'être coupés (algorithme 3), l'utilisation des projections permet une analyse différente du problème. On effectue pour chacune des projections d'objets appartenant à une sous-fenêtre de décomposition un balayage de tous les pixels situés dans la zone projetée (algorithme 4).

L'algorithme de tracé de rayons qui était :

Algorithme 3

Pour tous les pixels de la sous-fenêtre *faire*

Pour tous les objets de la sous-fenêtre *faire*

 Calculer l'intersection rayon-objet

Fin pour

 Afficher l'objet le plus proche

Fin pour

devient :

Algorithme 4

Pour tous les objets de la sous-fenêtre *faire*

Pour toute la surface délimitée par la projection de l'objet *faire*

 calculer intersection rayon-Liste_des_objets_possibles

 afficher l'objet le plus proche

Fin pour

Fin pour

On remarque que quand nous limitons le calcul d'intersections exclusivement aux zones de projection, nous ignorons naturellement les parties ne contenant aucune information. Le calcul d'intersections présenté pour l'algorithme 4 doit être effectué entre un rayon et tous les objets de la scène. En réalité, nous pouvons simplifier le nombre de comparaisons en retirant de la liste_des_objets_possibles la projection qui vient juste d'être étudiée. Considérons le simple exemple d'une scène contenant deux objets (figure 5).

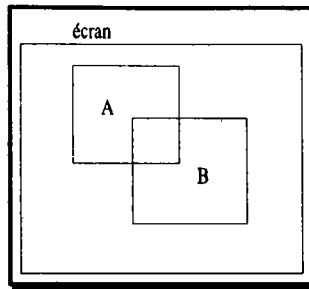


figure 5: scène projetée

Pour analyser complètement la scène nous considérons le premier élément de la liste_des_objets_possibles limitée, sur l'exemple, à deux éléments. Pour tous les pixels appartenant à la sous-fenêtre définie par la projection de A, nous cherchons une intersection entre un rayon passant par le centre du pixel et l'ensemble des éléments constituant la scène (algorithme 5). Une fois l'étude de A terminée nous avons étudié toutes les intersections de A et toutes les intersections des objets pouvant avoir une partie commune avec A. On peut retirer de la liste_des_objets_possibles l'élément A que l'on vient d'analyser complètement, plus aucun rayon primaire ne pour le couper.

Algorithme 5

Pour tous les pixels allant du coin haut gauche de A au coin bas droit de A *faire*

Si pixel non déjà étudié *Alors*

 rechercher une intersection avec A et B

Si intersections *alors* afficher la plus près

 marquer le pixel comme étant étudié

Fsi

Fsi

Fin pour

L'objet de la tête de liste étant totalement étudié, nous passons à l'élément suivant. Il ne reste, pour l'exemple, plus qu'un seul élément. Nous pouvons limiter la recherche d'intersection à l'objet B uniquement (algorithme 6).

Algorithme 6

Pour tous les pixels de B *faire*
 Si pixel non déjà étudié *alors*
 rechercher intersection avec B
 Si intersection *alors* afficher point
 marquer le point comme étudié
 Fsi
Fsi
Fpour

L'algorithme général (algorithme 7) parcourt et étudie récursivement toute la liste_des_objets_possibles. Pour chaque projection de l'élément de tête de la liste, il recherche une intersection entre un rayon et la liste toute entière.

Algorithme7

Pour tous les objets de la liste_objets_possibles *faire*
 Pour tous les pixels allant du coin haut gauche au coin bas droit de
 tête_de_liste(liste_objets_possibles) *faire*
 Si pixel non déjà-étudié *alors*
 rechercher intersections avec la liste_objets_possibles
 Fsi
 Si intersections *alors* afficher la plus près
 marquer le pixel comme déjà_étudié
 Fsi
 Fpour
 Passer à l'élément suivant de la liste
Fpour

La réalisation d'une maquette servant à valider les concepts de base d'une décomposition adaptative, nous a permis de déceler que l'ordre dans lequel les objets étaient traités provoquait des différences importantes sur le temps de calcul d'une image. Si les projections présentent de fortes disparités, il est important de toujours commencer par celle possédant la plus petite superficie et de traiter l'ensemble des projections dans l'ordre croissant de leur surface de projection. L'exemple de la figure 6 sert à mettre en évidence les variations parfois importantes du nombre de calculs nécessaires pour le traitement d'une scène. Les chiffres représentent la taille, en pixels, de l'écran et des différents objets.

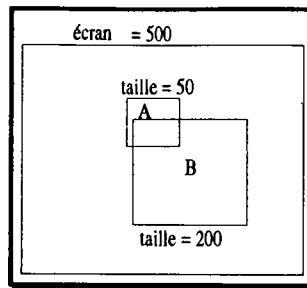


figure 6: influence de la taille

Si le premier objet étudié est B, on recherche les intersections entre A et B pour tous les pixels de la sous-fenêtre délimitée par la projection de B. On additionne ensuite le nombre d'intersections calculées pour la sous-fenêtre de A. Le nombre de calculs total d'intersections est de :

$$\text{Nb calculs d'intersections} = (200 * 200 * 2) + (50 * 50 * 1) = 80000 + 2500 = 82500$$

Par contre, si la première projection étudiée est A, le nombre de calcul est de :

$$\text{Nb calculs d'intersections} = (50 * 50 * 2) + (200 * 200 * 1) = 5000 + 40000 = 45000$$

Alors qu'un calcul conventionnel aurait donné comme valeur :

$$\text{Nb calculs d'intersections} = (500 * 500 * 2) = 500000$$

Ce simple exemple, montre l'influence de l'ordre dans lequel on étudie les différents objets. Il est important d'effectuer un tri croissant des surfaces de projection pour l'ensemble des éléments de la liste des objets et ceci pour chacune des sous-fenêtres de la décomposition.

Cette méthode de localisation est d'autant plus intéressante que le nombre d'objets dans la fenêtre étudiée reste modérément élevé (nous présentons au paragraphe 6 un procédé permettant de diminuer la complexité à l'intérieur des sous-fenêtres). Elle permet une application efficace de l'algorithme de tracé de rayons tant que le nombre d'objets reste inférieur à une valeur, que nous nomerons pour la suite de l'étude *nombre d'objets max*. Cette valeur arbitraire, pourra être modifiée avant chaque calcul par un opérateur en fonction de son expérience et du type de scène analysée. Elle est d'autant plus efficace que les éléments ont tendance à être concentrés avec des différences sensibles sur la taille des projections. Cependant elle n'est pas suffisante pour traiter efficacement les scènes comportant plusieurs dizaines d'objets par sous-fenêtre. C'est pourquoi nous avons associé à cette analyse une décomposition irrégulière qui tient compte des spécificités de l'approche et qui essaye, dans la mesure du possible, de se ramener dans l'une des situations favorables.

5. ADAPTABILITE A LA POSITION ET A LA FORME DES OBJETS: TRAITEMENT DES RAYONS SECONDAIRES ET D'OMBRAJE

L'obtention d'une image réaliste implique le traitement de rayons secondaires et d'ombrage. Nous présentons dans ce paragraphe deux points. Dans la présentation de l'extension des projections aux rayons secondaires (réfléchis et réfractés) nous expliquerons dans quelles limites la méthode peut être appliquée aux objets transparents et réfléchissants.

Nous verrons comment une simple adaptation de l'interprétation des données peut simplifier considérablement les tests de comparaison pour le calcul des ombres portées et des rayons secondaires.

5.1. Les rayons secondaires

Pour l'analyse des rayons réfractés et réfléchis, notre méthode s'apparente aux méthodes utilisées pour le beam-tracing (CH 2 § 4.2). Nous cherchons à propager, non pas un faisceau, mais une pyramide de visualisation dans laquelle nous pourrions réaliser les mêmes analyses que pour les rayons primaires. Comme pour le beam-tracing, notre approche connaît des limites liées au type des objets étudiés. Nous ne pouvons mettre en oeuvre le principe d'optimisation décrit, que si les objets étudiés sont approchés par des faces planes. La restriction est encore plus forte pour les objets transparents qui doivent obligatoirement avoir un coefficient de réfraction égale à 1.

Pour tous les autres cas, non pris en compte par notre analyse, nous adaptons l'algorithme de rendu à la fois aux formes et aux caractéristiques optiques des objets. Nous présenterons au paragraphe 8 comment nous modifions l'application des algorithmes en fonction du contenu de la scène. La méthode générale est un compromis de l'utilisation d'un ensemble de méthodes d'optimisation utilisant conjointement une décomposition adaptative et des méthodes plus conventionnelles comme les englobants.

5.1.1. Les rayons réfractés

Un des nombreux handicaps du beam-tracing est la limitation de la méthode aux objets modélisés par les facettes planes. La méthode proposée par Heckbert et Hanrahan [HEC 84] ne donne pas de solution générale simple pour le traitement des objets transparents. Les contraintes fortes imposées aux rayons et surfaces, font que le principe reste très limitatif dans son application.

Nous avons été confrontés aux mêmes problèmes et nous avons été obligés de restreindre le domaine d'application. La diffraction de la lumière n'étant pas linéaire pour les objets transparents, il est impossible d'assurer pour toutes les faces la constitution d'une pyramide de vision, où les quatre plans se rejoignent en un seul point. Il est en plus utopique de croire que les faces vont toujours être constituées de quatre côtés avec les bords parallèles deux à deux afin de permettre une propagation correcte du faisceau. Il est alors difficile de propager une zone de visualisation. La méthode d'optimisation que nous avons présentée n'est applicable qu'aux objets transparents ne faisant subir aucune déviation aux rayons lumineux (principalement les surfaces minces). A la différence du tracé de faisceaux, la forme de la face n'est pas, pour notre approche, un point bloquant. Il est toujours possible d'approcher la face plane par un rectangle qui l'englobe et de découper ensuite les zones non significatives.

Nous réalisons la projection de ce que voit l'observateur, à travers l'objet transparent, sur les différentes faces de l'objet en considérant chacune des faces comme un écran secondaire (figure 7).

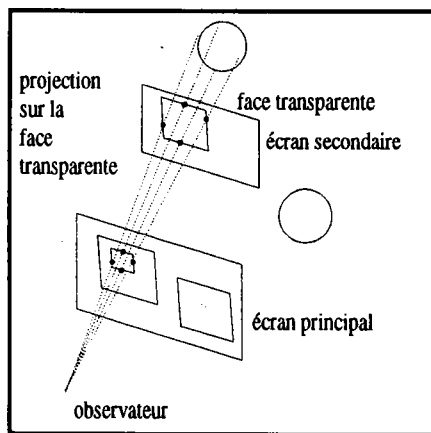


figure 7: projection sur les faces transparentes

On réalise ensuite, sur la face transparente, la même décomposition récursive effectuée sur l'écran principal.

Cette décomposition va permettre de limiter les tests de comparaison lors du calcul d'intersection entre un rayon et l'ensemble des objets. Lorsqu'un rayon (primaire ou secondaire) coupe une face transparente, on recherche la plus petite sous-fenêtre, de la décomposition associée à la face, contenant le point d'intersection (algorithme 8). On recherche alors une intersection entre le rayon, qui n'a subi aucune déviation, et la liste des objets associée à la sous-fenêtre (figure 8).

Algorithme 8

```

Recherche_intersection (rayon_courant, liste_objets_possibles)
Si intersection avec face_transparente alors
  Recherche_sous_fenêtre_minimale (point_intersection)
  Si sous_fenêtre_vide alors
    couleur_rayon = couleur_objet
  Sinon
    Rayon_refracté = rayon_courant
  Rechercher_intersection (rayon_réfléchi, liste_sous-fenêtre)

```

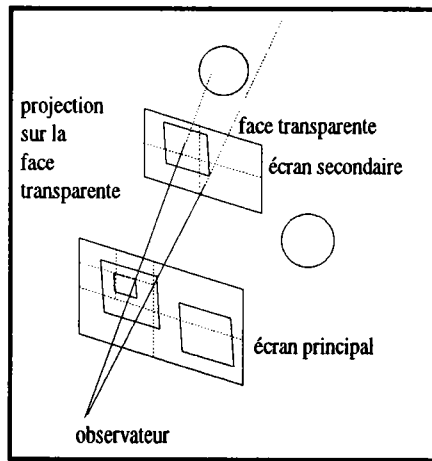


figure 8: analyse d'une face transparente

Comme l'observateur voit, à travers une face transparente, la même scène qu'il pourrait voir si cette face n'existait pas, on est amené à se poser la question de la nécessité de prendre en compte la présence de tels objets et pourquoi réaliser une projection supplémentaire sur la face étudiée plutôt que d'utiliser les projections effectuées sur l'écran principal (sur l'exemple de la figure 8). Premièrement nous sommes obligés de prendre en considération la couleur de l'objet transparent, les caractéristiques optiques ne se limitant pas à l'indice de réfraction du milieu traversé donc impossibilité d'éliminer de la scène ce type de face et d'objet. Deuxièmement, tous les objets transparents ne sont pas considérés comme n'ayant aucune épaisseur (vitre), il est possible que les objets visibles par chacune des faces transparentes d'un objet soient différents (cube). L'association d'une décomposition à chaque face transparente, permet souvent d'affiner l'analyse et de limiter le nombre de calculs d'intersections lors de la mise en oeuvre de l'algorithme du tracé de rayons. Il permet, par exemple, de ne pas prendre en compte les objets d'une même fenêtre ne se projetant pas sur la face transparente (figure 9).

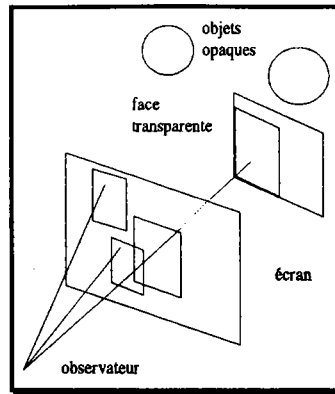


figure 9: projection sur une face transparente

On associe souvent, aux objets transparents, une caractéristique optique double. En plus de laisser passer les rayons lumineux, ils possèdent des propriétés de réflexion. Nous serons amené, suivant les caractéristiques des objets, à associer à une même face deux décompositions, une pour analyser la scène vue par transparence et une pour les objets vus par réflexion.

5.1.2. Les rayons réfléchis

Le traitement des faces spéculaires est effectué sur le même principe, si ce n'est que l'observateur secondaire est différent de l'observateur principal. On considère un observateur temporaire symétrique par rapport à la face réfléchissante (figure 10). La face, en cours d'analyse, devient un écran secondaire sur lequel on réalise la projection des objets vus par l'observateur secondaire. On réalise ensuite une décomposition récursive de l'espace de travail.

L'analyse s'apparente ensuite à celle effectuée pour les objets transparents (algorithme 9). Lorsque le rayon coupe une face réfléchissante, on recherche la plus petite subdivision contenant le point d'intersection. Si cette sous-fenêtre n'est pas vide, on calcule le rayon réfléchi et on recherche une intersection entre le rayon réfléchi et la liste associée à la sous-fenêtre. La restriction du domaine d'étude en effectuant une projection secondaire des objets, permet de diminuer les tests lors du calcul des intersections entre les rayons réfléchis et les objets de la scène.

Algorithme 9

Recherche_intersection (rayon_courant, liste_objets_possibles)

Si intersection avec face_réfléchissante **alors**

 Recherche_sous_fenêtre_minimale (point_intersection)

Si sous_fenêtre_vide **alors**

 couleur rayon = couleur objet

Sinon

 Calculer rayon_réfléchi

 Rechercher_intersection (rayon_réfléchi, liste_sous-fenêtre)

Fsi

Fsi

Si dans le champ de vision de l'observateur secondaire se trouve de nouveau un objet réfléchissant, l'analyse est reconduite jusqu'à ce que:

- il n'y ait plus dans le champ de vision que des objets opaques
- il n'y ait plus aucun objet
- la profondeur maximale de l'arbre d'analyse soit atteinte

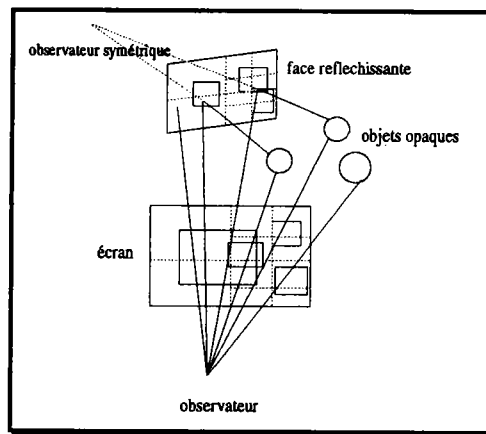


figure 10: traitement des faces réfléchissantes

5.2. Les rayons d'ombrage

5.2.1. L'éclairage direct

Nous prendrons comme convention, et ceci est vrai pour la majorité des algorithmes de tracé de rayons, qu'une source lumineuse est ponctuelle. Pour permettre d'adapter le principe de projection des objets, nous devons associer à chaque source une direction d'éclairage équivalente au point de mire pour un observateur. La recherche des objets éclairés est quelque peu différente de l'analyse des rayons primaires elle peut être comparée au traitement des faces visibles en assimilant la source à un observateur classique. Il suffit pour cela de joindre à la position de la source un plan simulant le plan écran d'un observateur.

Comme à priori on ne définit pas, pour une source lumineuse, une "pyramide d'éclairage" nous devons considérer l'ensemble des objets de la scène. Il suffit alors d'appliquer aux rayons d'ombrage le principe de décomposition utilisé pour les rayons primaires (figure 11).

On projette sur le plan de la source l'ensemble des objets ou des faces visibles depuis la source lumineuse. On définit un cadre, simulant l'écran, en prenant le valeur `coin_bas_gauche` et `coin_haut_droit` extrêmes des projections. On réalise ensuite une décomposition récursive de l'espace de projection.

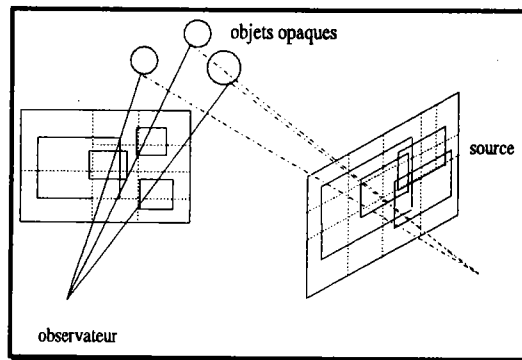


figure 11: décomposition pour les rayons d'ombrage

Pour savoir si un point d'intersection (que nous appellerons point origine) est ombré ou directement éclairé, il suffit de le projeter sur le plan de la lampe, de trouver la plus petite sous-fenêtre contenant le point projeté (algorithme 10). On cherche ensuite une intersection entre le rayon d'ombrage (passant par les deux points : point origine et source) et uniquement les éléments associés à la sous-fenêtre (figure 12). Si aucune intersection n'est trouvée, l'objet sera soumis à un éclairage direct. Sinon on compare les coordonnées du point d'origine avec tous les points d'intersection trouvés. Si aucune valeur n'est inférieure aux coordonnées du point origine, l'objet sera soumis à un éclairage direct sinon il se trouvera dans l'ombre portée d'un objet situé entre le point origine et la source lumineuse en tenant compte d'une éventuelle transparence de cet objet.

Algorithme 10

Pour toutes les lampes *faire*

 Projection du point d'intersection sur le plan de la lampe

 Rechercher la plus petite sous-fenêtre contenant le point

Si sous-fenêtre vide *alors*

 Objet directement éclairé

Sinon recherche intersection (rayon_ombrage, liste sous-fenêtre)

Si aucune intersection n'est plus proche que l'intersection d'origine *alors*
 point directement éclairé

Sinon point dans l'ombre

Fsi

Fsi

Fpour

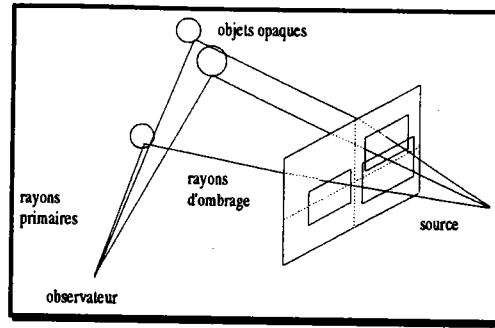


figure 12: rayons d'ombrage

Il est important de remarquer que la projection réalisée pour chacune des sources lumineuses est indépendante de la position de l'observateur. Cela permet, comme pour la radiosité, de ne pas avoir à recalculer la décomposition liée à chaque source lumineuse si entre deux images, la seule modification concerne l'observateur (position, point visé, focale).

5.2.2. L'éclairage indirect

L'une des carences importantes du tracé de rayons est la difficulté de prendre en compte les lumières indirectes qui proviennent essentiellement des objets réfléchissants (figure 13). Ceci est dû au fait qu'il est difficile de définir, pendant le suivi d'un rayon, quels vont être les objets pouvant contribuer à l'éclairage indirect de l'objet en cours d'étude. Sur l'exemple de la figure 13 l'observateur devrait voir dans le miroir un objet dont la face arrière est éclairée par une lumière se réfléchissant sur la surface spéculaire.

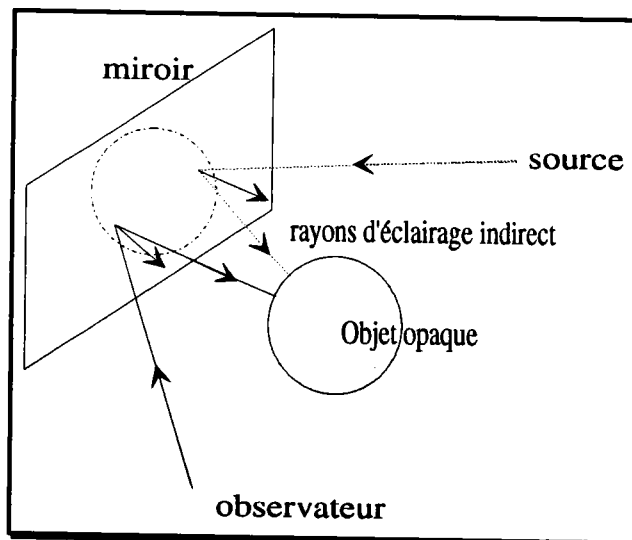


figure 13: éclairage indirect

L'analyse adaptative se déroulant en deux étapes distinctes, à savoir une étape de décomposition et une étape de calcul de rayons, permet, uniquement pour les miroirs plan, de traiter les sources indirectes relativement facilement. La démarche s'apparente au traitement des faces spéculaires (paragraphe 3.1.2). En effet lors de la phase de calcul des projections et des décompositions nous créons, pour les miroirs, un ensemble d'observateurs secondaires qui peuvent avoir dans leur champ de vision une source lumineuse. Dans ce cas, une certaine quantité de lumière va être renvoyée par la face réfléchissante et viendra éclairer indirectement les objets de la scène. Nous pouvons créer une source secondaire fictive qui se situe symétriquement par rapport à la face réfléchissante étudiée (figure 14). Nous effectuons ensuite, comme pour une source réelle, une projection et une décomposition des objets éclairés par la source symétrique. Le plan du miroir devient le plan de projection, et ses limites (ramenée à un parallélépipède rectangle) définissent les limites factives de la zone éclairée.

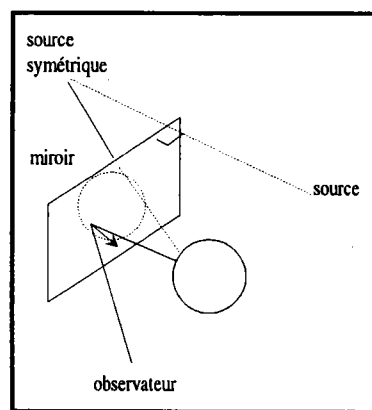


figure 14: création d'une source secondaire

Il convient alors d'associer à la face réfléchissante la liste des sources lumineuses secondaires correspondantes. On ne peut considérer la source lumineuse secondaire comme une source normale. Comme le montre la figure 15 la source symétrique ne doit être prise en compte que pour les rayons réfléchis par le miroir. La considérer comme une source globale engendrerait des erreurs sur l'éclairage des objets ne se réfléchissant pas sur la face spéculaire. En effet, on ajouterait sur ces objets une quantité de lumière provenant d'une source qui pour eux, n'existe pas.

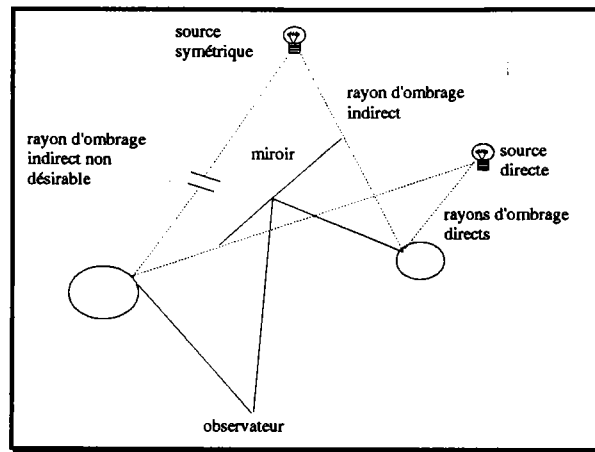


figure 15: éclairage indirect

La méthode de décomposition adaptative appliquée aux rayons d'ombrage permet de traiter efficacement trois problèmes importants du tracé de rayons. Le premier est l'optimisation du calcul des rayons d'ombrage par une diminution des tests d'intersection. Le second est l'indépendance de l'analyse par rapport à l'observateur. La décomposition reste valide entre deux calculs si le seul déplacement est celui de l'observateur. Le troisième est la possibilité de prendre en compte simplement les éclairages indirects.

5.3. Méthode générale

La méthode que nous venons de décrire repose sur deux phases bien distinctes. La première consiste à réaliser une pré-analyse du contenu de la scène, en associant aux faces planes non opaques une décomposition. Comme pour le beam-tracing on propage un cône de visualisation, ou plus exactement, on effectue une propagation des projections. Une même faces peut, à des profondeurs d'étude différentes, se voir associer plusieurs décomposition. La seconde phase consiste à appliquer un algorithme de tracé de rayons en adaptant le type d'optimisation (décomposition ou englobants), à la forme et aux caractéristiques optiques de l'objet. La figure 16 montre trois décompositions, une sur l'écran principal pour l'optimisation des rayons primaires, une autre sur une face réfléchissante pour certains rayons secondaires enfin une sur la source lumineuse pour les rayons d'ombrage.

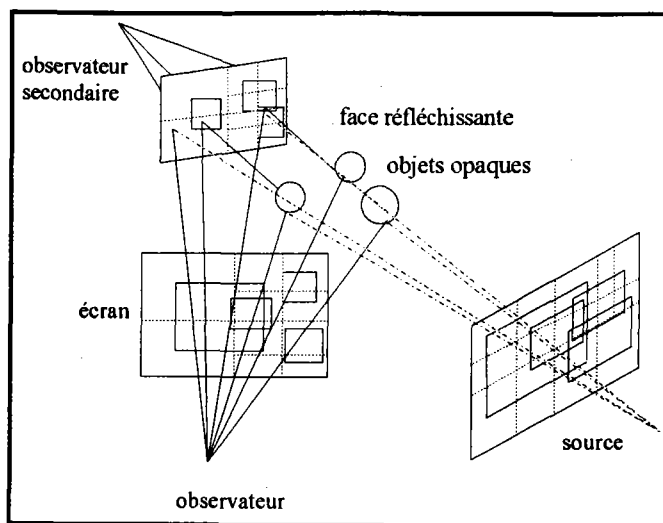


figure 16: propagation des rayons

Si la localisation des objets, par projection des englobants, nous renseigne sur la position des objets et permet de limiter, à certaines zones, l'application des algorithmes de rendu, nous avons souhaiter définir une méthode de décomposition capable d'exploiter l'ensemble de ces informations et de s'adapter au contenu de la scène.

6. ADAPTABILITE AU CONTENU DE LA SCENE

Si l'on se remémore les limites des décompositions classiques, on remarque que les subdivisions de scènes ne servent qu'à diminuer de la complexité à l'intérieur des sous-fenêtres. Le traitement ne s'adapte aucunement au contenu. La limitation des tests d'arrêt mis en oeuvre, à savoir fenêtre vide, fenêtre recouverte, élément unique et fenêtre minimale (souvent le pixel), contribue souvent à la création de situations paradoxales. Si l'on considère l'exemple de la figure 17, une décomposition classique entraîne une subdivision abusive, qui, en plus d'être coûteuse en temps de traitement et en place mémoire, n'apportera aucune information qui permettrait d'aborder le calcul plus simplement.

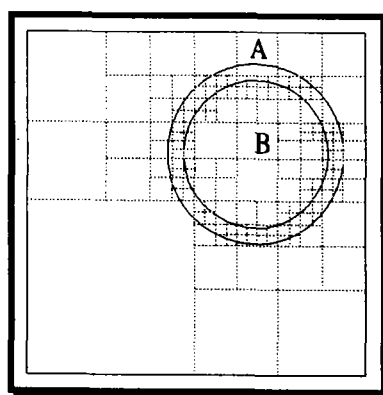


figure 17: décomposition excessive

Un autre point fort de notre approche est que nous avons mis en oeuvre une décomposition active qui est en mesure de prendre en compte un certain nombre de paramètres qui vont orienter ou non une décomposition vers l'arrêt. Ils vont permettre de guider la subdivision de l'espace de travail pour mettre en évidence des situations que nous savons bien traiter. Nous avons tiré parti du fait que nous étions en mesure d'avoir une application différente de l'algorithme du tracé de rayons. Nous avons introduit dans la définition des tests d'arrêt, des critères de situation et d'occupation d'espace. Nous avons pour cela constitué deux catégories de tests, à savoir des tests que nous appellerons *terminaux* qui ont pour rôle d'arrêter la décomposition et des tests *d'orientation*, qui vont orienter la décision vers l'arrêt ou la poursuite de la décomposition.

6.1. Les tests d'arrêt terminaux

Les tests terminaux vont avoir pour rôle d'arrêter une décomposition et cela indépendamment des autres tests. Classiquement par rapport aux autres algorithmes de décomposition, nous avons retenu trois tests, à savoir ,fenêtre vide, fenêtre ne contenant qu'un seul élément et taille minimale de fenêtre atteinte. Aucune de ces situations ne sera explicitement recherchée, c'est-à-dire que ce ne sont pas elles qui guident la décomposition. Elles seront uniquement détectées lorsqu'elles se présenteront. Nous n'avons pas retenu le test fenêtre_recouverte, un des tests d'orientation que nous avons définis aura entre autres cette fonctionnalité.

6.2. Les tests d'orientation

La deuxième catégorie de tests que nous avons définie et appelée, *tests d'orientation*, sert à acquérir un ensemble d'informations sur la répartition des objets dans la scène. Ces tests vont servir à pallier les carences des tests d'arrêt et notamment au problème de la décomposition abusive. Pour cela, nous implantons trois tests supplémentaires qui vont analyser la concentration des objets, la possibilité d'effectuer un calcul de tracé de rayons en fonction du nombre d'objets et enfin, un test qui déterminera si une décomposition apporterait ou non une simplification de l'analyse du contenu.

6.2.1. Les tests d'orientation : le taux d'occupation

6.2.1.1. définition du test

Généralement les décompositions ne tiennent pas compte de la topologie de la scène. C'est pourquoi on arrive encore à des situations paradoxales comme celle décrite en 6 (figure 17) où une décomposition, qui atteint des niveaux importants, n'est aucunement représentative du contenu de la scène.

Si l'on était en mesure de tirer certaines informations sur la position relative des objets on pourrait éviter bon nombre de traitements inutiles. C'est pourquoi nous avons défini un test qui nous apporte des informations sur la surface occupée en fonction du nombre d'objets : le taux d'occupation. On appelle taux d'occupation le rapport entre la taille de la fenêtre étudiée et la surface occupée par les objets à l'intérieur de cette fenêtre.

Pour obtenir la valeur exacte du taux d'occupation, il faut additionner les surfaces des projections et soustraire les surfaces des intersections. Une première maquette a montré que ce calcul est très coûteux en temps de traitement, on le simplifie en considérant que la surface totale est donnée en prenant les valeurs minimales et maximales des coins bas gauche et haut droit de toutes les projections (figure 18).

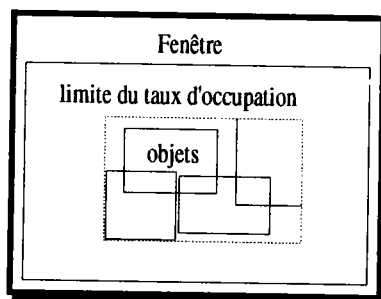


figure 18: approximation du taux d'occupation

Il est évident qu'il existe des situations où la valeur du taux d'occupation ne sera pas représentative de la situation réelle (figure 20). Nous associerons au test du taux d'occupation le nombre d'objets contenus dans la sous-fenêtre afin de minimiser les situations pouvant engendrer des erreurs d'interprétation.

6.2.1.2. Application du test

Ce test donne des informations sur la concentration des objets à l'intérieur d'une fenêtre. En fonction de la valeur du taux et du nombre d'objets, nous sommes en mesure d'interpréter un ensemble de situations et de prendre ou non la décision de poursuivre la décomposition.

* Si la valeur calculée indique un taux d'occupation faible, on interprète le résultat comme étant représentatif de la concentration de plusieurs objets avec une forte probabilité que ceux-ci s'intersectent (figure 19).

Une décomposition classique imposerait systématiquement la poursuite de la subdivision et on atteindrait obligatoirement le niveau du pixel pour toutes les zones comportant des intersections. Avec l'approche que nous avons définie, la détection de cette situation marque une suspension dans l'analyse. Ce test seul ne peut décider de la suite du traitement. Nous regardons ensuite le nombre d'objets. S'il est inférieur au *nombre d'objets max* on se trouve dans un cas simple, savons bien le traiter (voir paragraphe 4), la subdivision est arrêtée. Si le nombre d'objets est plus important, aller un niveau plus bas dans la décomposition simplifiera les tests du calcul d'intersections.

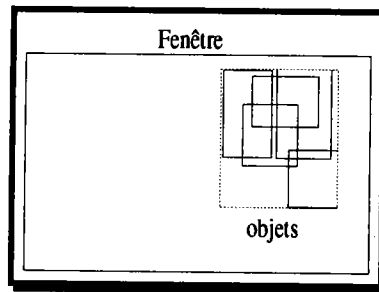


figure 19: concentration des objets

* Si le taux d'occupation est important.

Un taux d'occupation important ne peut être représentatif du contenu de la scène. Il existe en effet des scènes très différentes pouvant présenter des taux voisins (figure 20). Il existe des cas défavorables où l'interprétation des données s'avère fausse.

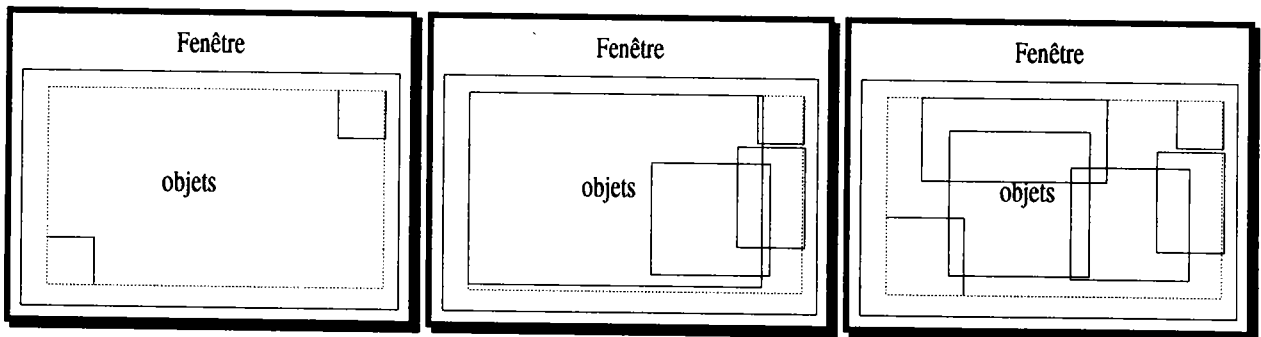


figure 20: taux d'occupation important

Dans tous les cas de figure, si le taux d'occupation est important, on a intérêt à décomposer la fenêtre d'étude. En effet, si le nombre d'éléments est faible on se trouve en présence de trois situations possibles :

- les éléments à l'intérieur de la fenêtre sont petits, dans ce cas une décomposition isolera les objets et simplifiera le traitement (figure 20 dessin de gauche).
- un élément possède une taille dominante par rapport aux autres objets, dans ce cas une décomposition isole des grandes parties simples à traiter (figure 20 dessin du centre).

- les éléments sont de taille quelconque, c'est simplement leur répartition qui a induit cette situation. Dans ce cas, si le nombre d'objets est proche de la valeur *nombre d'objets max*, la décomposition n'apportera pas une simplification significative de la scène sauf, si la fenêtre étudiée est très supérieure à la fenêtre minimale. Par contre si le nombre d'objets est élevé, on choisira de poursuivre la décomposition (figure 20 dessin de droite). L'algorithme définissant le test du taux d'occupation s'écrit :

Algorithme 11

Si le taux d'occupation est élevé **Alors**

Si le nombre d'objets est proche de *nb objet max* **Alors** arrêt

Sinon Si la taille de la fenêtre est proche de la fenêtre minimale **Alors** arrêt

Sinon poursuivre la décomposition

Fsi

Fsi

Fsi

Dans bien des situations ce test se combine avec les valeurs de *fenêtre minimale* (test terminal) et *nb objets maximal* que nous présentons maintenant.

6.2.2. les tests d'orientation : le nombre d'objets maximal

Le principe retenu pour analyser le contenu d'une sous-fenêtre permet de traiter des subdivisions de scène contenant plus d'un élément. Une analyse préalable réalisée sur une maquette a montré que l'on pouvait fixer le nombre maximal d'éléments par fenêtre, non plus à un comme dans les décompositions classiques, mais à des valeurs supérieures (4 à 7 éléments sur les scènes analysées). Nous avons constaté qu'un grand nombre d'objets n'était pas pénalisant, si la taille en pixels de l'image, est relativement faible (< 200 par 200), le coût engendré par des décompositions supplémentaires ne compensait pas le gain de temps au niveau du calcul des rayons. Par contre nous avons intérêt à diminuer cette valeur si la taille en pixels devenait grande. Cependant, le test doit être associé au taux d'occupation si la taille de la fenêtre étudiée est importante. En effet, si pour une fenêtre de taille importante (500*500) le nombre d'objets est peu élevé et que le taux d'occupation est important, on a tout intérêt à décomposer pour isoler les éléments. Nous expliciterons plus en détail au paragraphe 7.2 la nécessité d'utiliser ce test en fonction des situations rencontrées.

6.2.3. les tests d'orientation : égalité de contenu entre les sous-fenêtres

Le dernier test que nous utilisons, permet d'approfondir nos connaissances sur la position des objets à l'intérieur de la scène. Il nous renseigne sur d'éventuelles intersections entre les objets. Pour ce faire, nous recherchons quels sont les objets coupés par un rayon lancé au centre de la scène (figure 21). Si le rayon central coupe tous les éléments de la scène, ceux-ci vont se

retrouver à nouveau dans les différentes sous-fenêtres. Nous pouvons alors affirmer que le contenu des subdivisions peut être de même complexité que la fenêtre mère, le problème étant simplement déplacé à un niveau plus bas. Il convient de faire intervenir un test que nous avons déjà défini: le taux d'occupation.

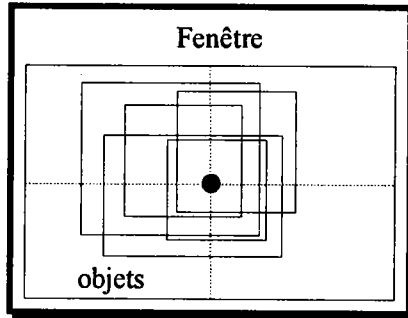


figure 21: égalité du contenu des sous-fenêtres

Si le taux d'occupation est important, la surface commune aux différentes projections pourra être considérée comme peu importante (figure 22), elle risque même dans le pire des cas, d'être réduite à un simple point.,

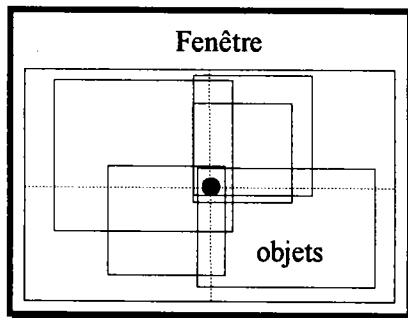


figure 22: taux d'occupation important

Cette situation aura tendance à demander la poursuite de la décomposition afin de simplifier le contenu des différentes sous-fenêtres. Par contre, si le taux d'occupation est faible (figure 23), si le nombre d'objets est proche de la valeur *nb objets max* et si la taille de la fenêtre n'est pas proche de la taille *fenêtre minimale* alors la subdivision devra être poursuivie.

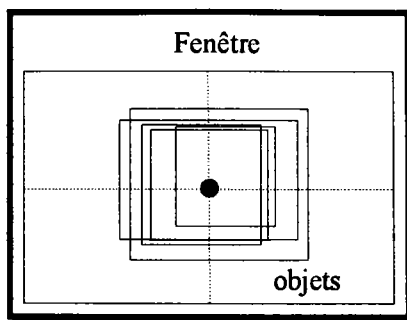


figure 23: taux d'occupation faible

L'algorithme définissant ce test s'écrit :

Algorithme 12

Si égalité entre racine et point central *alors*

Si condition taille fenêtre et nb objets max le permettent *alors*

Si taux d'occupation élevé Alors poursuite décomposition

Sinon arrêt

Fsi

Fsi

Fsi

7. ADAPTABILITE DES TESTS ET DES VALEURS A L'UTILISATEUR.

Nous avons défini une méthode d'optimisation en réalisant une séparation entre le traitement de la décomposition et la gestion des tests d'arrêt. Cette division a montré un intérêt indéniable aussi bien au niveau des résultats qu'au niveau souplesse d'utilisation. Nous avons laissé la possibilité à l'utilisateur de jouer sur l'ensemble des paramètres liés aux tests d'arrêt. Il peut, en effet, modifier avant chaque calcul d'image, le nombre de tests d'arrêt mis en oeuvre ainsi que leur valeur. L'opérateur peut ainsi sélectionner, selon son expérience, les tests les plus appropriés à la scène analysée. Il peut combiner, simplement en les sélectionnant, les tests de: *fenêtre vide*, *objet unique*, *taille minimale*, *taux d'occupation* et *nombre d'objets*. Il peut modifier la valeur des constantes de *taille minimale en pixels*, *le nombre d'objets maximum par fenêtre* et *taux d'occupation*. Nous avons aussi laissé à l'opérateur le choix des principes d'optimisation mis en oeuvre. Il a le choix, pour les rayons primaires, secondaires et d'ombrage, entre plusieurs types d'optimisations: englobants, Lightbuffer et décomposition. Il peut, pour chaque type de rayons, choisir un type d'optimisation différent. Certaines combinaisons semblent cependant non appropriées. Il est algorithmiquement difficile de choisir une optimisation de type décomposition pour les rayons secondaires alors que l'on a choisi une optimisation de type englobant pour les rayons primaires. Seuls les rayons d'ombrage sont véritablement indépendants des autres types de rayons.

Lors de la validation d'une maquette de pré-étude nous avons constaté que selon les caractéristiques de la scène, les combinaisons des tests effectuées peuvent influencer considérablement les performances. La présence de tous les tests n'étant pas nécessaire en même temps, il serait intéressant d'être en mesure de déduire d'une analyse de la scène, les meilleurs compromis au niveau de la décomposition. L'idéale serait une déduction automatique qui générerait, un peu comme un système expert, l'organisation des tests de décomposition. Pour cela, nous avons cherché à extraire d'une scène, les caractéristiques qui influent sur les performances de calcul et nous avons essayé d'associer à chaque type de scène, les tests d'arrêt les plus performants pour la décomposition.

7.1. Pour les images de petite taille.

Comme pour toutes les méthodes d'optimisation basées sur une analyse de contenu de la scène, les temps de prétraitement nécessaires à la mise en oeuvre de la décomposition adaptative ne sont plus négligeables pour tout un ensemble d'images. Plus précisément, de la taille de l'image calculée vont dépendre les performances, donc indirectement, la nécessité d'appliquer ou non tout un environnement d'analyse coûteux en place mémoire et en temps machine. Afin que la méthode de décomposition ne devienne pas pénalisante pour le calcul, nous avons tout intérêt, pour certaines images de petite taille, à mettre en oeuvre un algorithme de tracé de rayons ayant comme seule optimisation une approximation des objets par des englobants. Nous devons toutefois tenir compte, même pour des fenêtres de petite dimension, de situations où une décomposition s'avère utile. En effet, si le nombre d'objets traités est important, une décomposition diminuera le nombre de comparaisons nécessaires au calcul de chacun des rayons. Un autre critère entre en ligne de compte pour le choix des optimisations: la répartition des objets à l'intérieur de la fenêtre. Une analyse du taux d'occupation apporte des informations importantes sur les possibilités d'intersection d'objets. Si l'ensemble des éléments est concentré sur une petite zone de l'image alors il risque d'apparaître des zones vides d'information pour lesquelles le calcul des rayons devient inutile. Nous avons, pour des fenêtres de petite taille un choix de méthode d'optimisation tenant compte aussi bien du nombre d'objets et de leur répartition dans la scène. Le pseudo algorithme présenté ci-dessous (algorithme 13) montre les choix retenus pour la type de l'optimisation et pour les tests d'arrêt à mettre en oeuvre.

Algorithme 13

Si la taille de la fenêtre est inférieure à une taille critique **Alors**

Si le nombre d'objets est inférieur au *nombre_d'objets_max* **Alors**
 utiliser les englobants comme unique forme d'optimisation

Sinon Si le taux d'occupation est important **Alors**
 utiliser une optimisation par décomposition adaptative

Sinon utiliser les englobants comme unique forme d'optimisation

Pour des images de petite taille, deux critères vont influencer sur le choix de la méthode d'optimisation mise en oeuvre. Le premier de ces critères est le nombre d'objets. Suivant sa valeur on optera soit pour une optimisation par les englobants soit par décomposition. Le deuxième de ces critères est le taux d'occupation. Un taux de concentration faible indiquera la présence de zones où les rayons ne couperont aucun objet. Pour les régions vides des petites fenêtres il sera plus intéressant de rechercher des intersections avec uniquement des englobants plutôt que de rechercher à isoler, avec à une décomposition coûteuse en temps et en place, des zones qui ne représentent finalement que très peu de pixels.

7.2. Pour les images de grande taille.

Suivant les situations rencontrées, l'application de tous les tests d'arrêt à toutes les phases de la décomposition n'est pas obligatoire. Il est en effet possible, dans certains cas, de ne plus prendre en compte les tests qui deviennent inopérant ou de modifier les arrangements entre les tests utilisés.

A la première analyse du contenu par la méthode de décomposition adaptative, tous les tests que nous avons présentés au paragraphe 6 doivent être pris en compte. Suivant les informations recueillies, nous pouvons orienter et adapter l'analyse qui sera effectuée sur les sous-fenêtres issues de la subdivision de l'espace initial.

7.2.1. Le test du nombre d'objets maximum.

Pour certains types de scène nous pouvons au cours de la subdivision ignorer le test du nombre d'objet maximum. Se plaçant toujours dans une situation où la taille de l'image à calculer est grande, si le nombre d'objets est inférieur au *nb_objets_max* nous pouvons, par une analyse du taux d'occupation, avoir une image de la répartition des objets à l'intérieur de l'espace. Si cette valeur est importante, on peut espérer que les éléments se disséminent sur l'ensemble de la fenêtre. Dans ce cas une décomposition peut isoler une partie des éléments. Pour l'analyse des sous-fenêtres, le test du *nb_objets_max* devient obsolète. Déjà, à un niveau supérieur, le nombre d'éléments de la scène était inférieur à cette valeur, il le sera obligatoirement pour l'étude des différentes sous-fenêtres. La réalisation d'une maquette a montré que si la décomposition entraîne la création de subdivisions dont la taille en pixels approche d'une valeur critique, d'environ 200 par 200 pixels, le test du nombre d'objets minimum doit de nouveau être pris en compte. En effet, pour des fenêtres approchant ces valeurs, l'étude des objets par la méthode que nous avons décrite au paragraphe 4 devient plus performante, les coûts engendrés par la décomposition ont tendance à pénaliser les temps de calcul.

7.2.2. Le taux d'occupation.

Nous pouvons avoir le même type d'approche avec le test du taux d'occupation. Si la fenêtre étudiée contient un nombre élevé d'objets et si le taux d'occupation est très important ($> 80\%$) une décomposition va obligatoirement créer des subdivisions où le taux d'occupation va rester très fort. L'interprétation du test ne sera plus significative. Nous pourrions sortir temporairement le taux d'occupation de la liste des tests d'orientation prioritaires. Cependant, comme pour le test du nombre d'objets maximum, lorsque la sous-fenêtre atteint des dimensions plus modestes le taux d'occupation redevient un test qu'il faudra prendre en compte afin d'éviter une décomposition abusive de l'espace de travail.

8. ADAPTABILITE DES ALGORITHMES AUX OBJETS.

Toutes les scènes utilisées pour tester notre algorithme d'optimisation ont été développées sous le système de C.A.O. S.A.C.A.D.O., Système Adaptatif de Conception et d'Assistance au Développement par Ordinateur qui s'appuie sur deux concepts de base: convivialité et ouverture. La convivialité repose sur un dialogue souple, qui permet à l'utilisateur de construire des objets avec un minimum de contraintes, et cela, sans faire appel à un langage informatique. De plus, l'ouverture lui permet de définir un dialogue qui lui soit adapté, c'est-à-dire correspondant le mieux au contexte dans lequel il évolue.

Actuellement SACADO propose une version 2D et une version 3D s'appuyant sur un modèle par les limites dont le constructeur de base est l'extrusion. Pour apporter plus de réalisme aux images, dont les objets sont modélisés par les limites, nous avons développé au laboratoire un algorithme de lissage basé sur une interpolation des normales donnant un résultat équivalent au lissage de Phong [LAN 91].

Nous avons présenté aux paragraphes précédents une méthode d'optimisation qui ne pouvait être appliquée qu'aux objets modélisés par des faces planes. Pour mettre en oeuvre un algorithme général, il a fallu adapter les optimisations aux caractéristiques des objets.

Dans la majorité des systèmes de synthèse d'images, la même technique de décomposition sert à calculer toutes les scènes. Or, une scène peut être constituée d'objets de tous types: faces planes, surfaces gauches, quadriques. Une seule technique d'optimisation du tracé de rayons n'est pas toujours adaptée à tous ces types de modélisation. Plutôt que de définir une méthode générale, nous avons préféré opter pour une analyse différente en fonction de la forme et des caractéristiques optiques de l'objet.

Pour permettre l'affichage d'objets quelconques il fallait que les algorithmes puissent s'adapter au contenu de la scène. La méthode d'optimisation que nous avons définie ne pouvant s'appliquer qu'aux surfaces planes, il a fallu définir un algorithme capable d'adapter les calculs à l'objet étudié. Cette adaptabilité porte à la fois sur les caractéristiques optiques de l'objet et sur les caractéristiques géométriques.

On utilisera par exemple l'optimisation que nous avons définie pour traiter les objets à faces planes et nous passerons à une optimisation plus conventionnelle, comme les englobants sphériques par

exemple, quand nous analyserons tout autre type d'objets. Cette notion d'adapter les algorithmes aux types des objets n'est pas obligatoirement nouvelle. Déjà en 82, mais pour une application différente, Crow [CRO 82] propose une méthode qui en deux étapes détecte des groupes d'objets indépendants et leur affecte une priorité d'affichage. Pour chacun des groupes il utilise l'algorithme de visibilité le plus approprié. On retrouve aussi une notion d'adaptabilité dans les travaux de Jahami [JAH 91] qui utilise un mixage des algorithmes de tracé de rayons et d'Atherton pour visualiser une scène avec la méthode la plus adaptée. Il calcule en fait séparément deux images. La première, rangée dans la base TR, ne prend en compte que les objets ne pouvant être traités que par le tracé de rayons: les objets réfléchissants, transparents et de révolution. Pour réaliser ce premier calcul, la méthode est obligée de tenir compte de tous les éléments.

La seconde image, rangée dans la base AT, ne représente que les objets analysables par l'algorithme d'Atherton, à savoir les objets opaques à faces planes. L'image finale est le résultat de la fusion des deux bases d'images. La méthode, très intéressante dans le principe, traite très difficilement les problèmes d'interaction entre les réflexions, les réfractions et les ombres portées. L'éclatement de la scène en plusieurs bases pour lesquelles on utilise des algorithmes différents, rend difficile la prise en compte de contraintes entre objets appartenant à des bases différentes. C'est pourquoi nous avons préféré utiliser un unique algorithme de rendu, à savoir le tracé de rayons, que nous appliquons avec des méthodes d'optimisations différentes en fonction des caractéristiques optiques et géométriques des objets.

8.1. Adaptabilité aux caractéristiques optiques des objets.

Indépendamment des contraintes liées à la forme, la méthode d'optimisation que nous avons présentée ne pouvait être appliquée aux objets transparents que si ceux-ci avaient un coefficient de réfraction égal à un. Pour toutes les autres formes de réfraction nous devons adapter les optimisations. Dès qu'un rayon coupe un objet transparent ne correspondant pas aux caractéristiques d'une optimisation par décomposition, on considérera que la forme d'optimisation des rayons secondaires devient, et ceci simplement pour l'objet en cours d'étude, les englobants (figure 24).

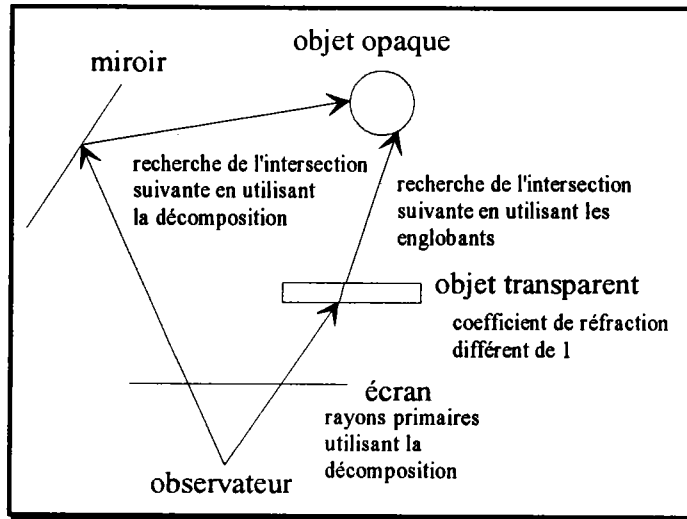


figure 24: modification du type d'optimisation

Lorsque l'on réalise une modification de la méthode d'optimisation, elle doit être conservée pour toute l'étude du rayon même si celui-ci coupe, par la suite, un objet pour lequel on a retenu une optimisation par décomposition. En effet, une décomposition sur les faces est effectuée soit pour l'observateur principal soit pour un observateur secondaire, l'orientation du rayon étudié ne correspond pas, sauf cas particuliers, à la direction qui a servi à la décomposition (figure 25).

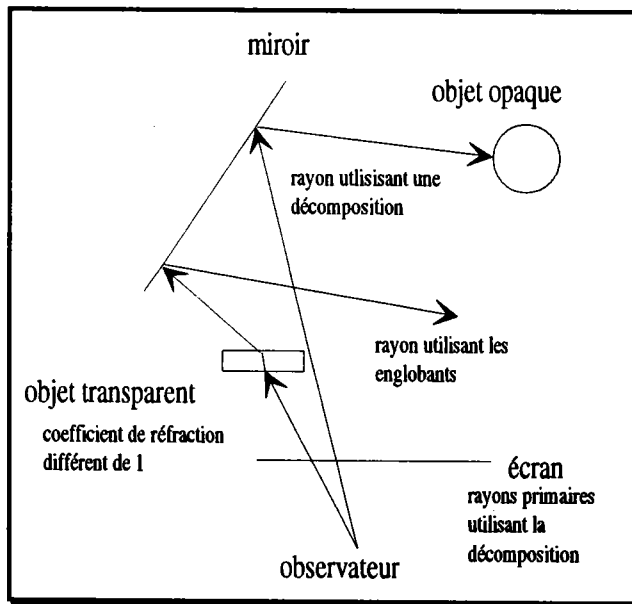


figure 25: adaptabilité de l'optimisation au rayon

Nous venons de présenter de quelle manière nous apportons une solution au traitement des objets transparents ayant un coefficient de réfraction différent de un. L'adaptation aux caractéristiques optiques est possible car l'algorithme de rendu reste identique quel que soit le genre de la décomposition mis en oeuvre. Il reste à présenter de quelle manière nous pouvons adapter les optimisations aux caractéristiques géométriques des objets.

8.2. Adaptabilité aux caractéristiques géométriques des objets.

Dans la mesure où aucune méthode de modélisation ne permet de représenter simplement et complètement tous les types d'objet, il serait intéressant de pouvoir faire cohabiter au sein d'un même système des objets modélisés aussi bien par ses limites que par leurs équations mathématiques. Actuellement, seul un algorithme de tracé de rayons, avec des adaptations pour les modèles CSG (ray-casting) permet de calculer l'illumination globale d'une scène, avec des effets optiques complexes, sans modification préalable de la représentation de l'objet. Nécessitant simplement comme information les coordonnées du point d'intersection et la normale en ce point, il autorise la combinaison de plusieurs formes de représentations des objets au sein d'une même scène. Il suffit d'adapter l'algorithme d'intersection au mode de description de l'objet, ce qui bien souvent le plus difficile.

Dans un souci d'homogénéité et d'amélioration des temps de calcul, nous avons choisi d'adapter les méthodes d'optimisation aux objets, tout en conservant comme algorithme d'affichage l'algorithme de tracé de rayons. D'une manière générale, on cherche à associer automatiquement la forme d'optimisation la plus adaptée à la définition des objets.

Lorsque l'objet analysé est constitué de faces planes, nous pouvons mettre en oeuvre la méthode de décomposition adaptative que nous avons définie. Par contre lorsqu'un objet possède des faces gauches la normale à la face n'est plus unique, il devient possible de trouver un unique observateur symétrique (figure 26)

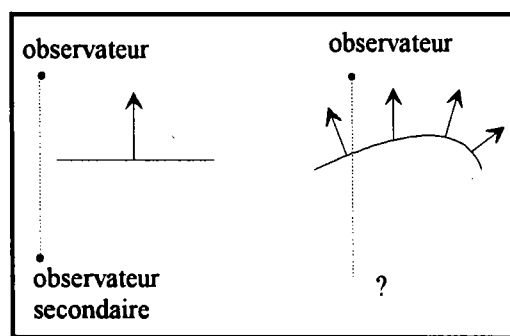


figure 26: recherche d'un observateur secondaire

La méthode de lissage développée au laboratoire simule, à partir d'une face plane, la courbure de la surface en interpolant les normales aux sommets (figure 27). Nous devons donc gérer, au sein d'une même scène, des objets constitués exclusivement de faces planes et de "faux" objets de révolution.

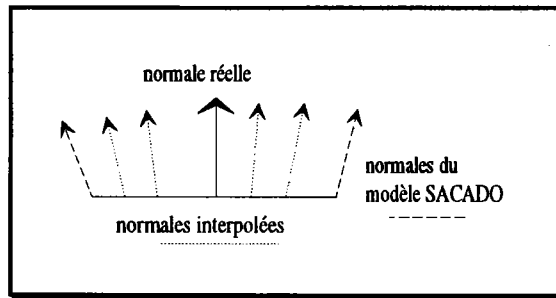


figure 27: interpolation des normales

Pour traiter les objets comportant des faces non planes nous avons la même démarche que pour les objets transparents. Lorsque le rayon coupe un objet réfléchissant sur lequel la décomposition adaptative ne peut être appliquée, nous choisissons, pour calculer les rayons secondaires une méthode d'optimisation utilisant des englobants, et ceci pour l'analyse complète de l'objet, (figure 28).

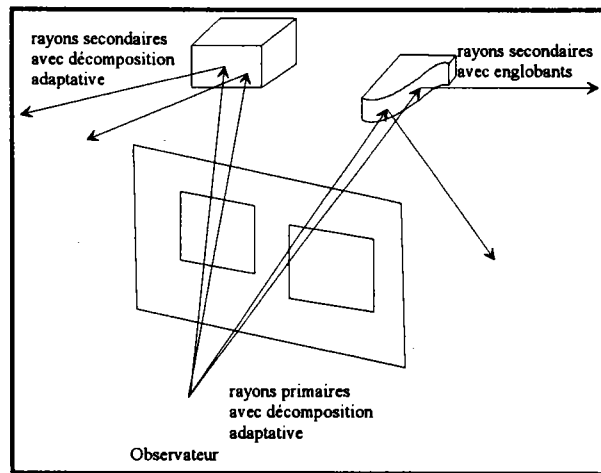


figure 28: adaptabilité à la forme de l'objet

9. CONCLUSION

Le tracé de rayons n'est qu'une méthode parmi d'autres pour la visualisation en CAO. On peut souhaiter mettre à la disposition de l'opérateur un certain nombre de méthodes plus ou moins rapides et donnant des résultats de plus ou moins bonne qualité (ne serait-ce qu'en limitant la profondeur de l'arbre de rayons dans le tracé de rayons). Les systèmes de CAO devraient être capable d'adapter leurs méthodes de visualisation en fonction des caractéristiques de la scène, de la qualité des résultats et des temps de réponse souhaités : cette adaptabilité passe, tout en utilisant la décomposition, par une mise en oeuvre conjointe ou pas de plusieurs types d'algorithme.

La méthode que nous avons présentée n'a pas la prétention de résoudre tous les problèmes liés à la cohabitation de plusieurs algorithmes au sein d'un même système de visualisation. Cependant nous apportons un certain nombre de solutions permettant de définir un outil souple et convivial. Pour que le système soit le plus généraliste possible nous avons souhaité approfondir d'une part l'analyse du contenu de la scène et d'autre part l'adaptabilité des algorithmes à la modélisation des objets.

Ces deux notions semblent fondamentales à un système ouvert, capable d'adapter ses algorithmes de visualisation à la scène étudiée en tenant compte à la fois des désirs de l'opérateur. Nous avons intégré ces deux idées aux différentes étapes de la décomposition adaptative.

La première de ces étapes consiste en une pré-analyse de contenu basée sur la recherche de propagation d'une pyramide de visualisation. A l'intérieur de ces zones une analyse du contenu guide une décomposition récursive dont le but est de diminuer localement la complexité de la scène. Cette analyse prend en compte plusieurs paramètres liés au nombre d'objets, à leur répartition dans la scène et à la probabilité que ceux-ci ont de se couper. L'élargissement de l'éventail des caractéristiques analysées permet d'affiner les connaissances du contenu et d'adapter les subdivisions pour que celles-ci puissent être le plus optimales possible.

La seconde étape consiste en une application selective du tracé de rayons. L'utilisation des projections d'objets permet de limiter le calcul d'intersections uniquement aux zones susceptibles de contenir de l'information. De plus, nous apportons une solution au problème principal de la propagation d'un cône de visualisation, que l'on retrouve classiquement dans le beam tracing, en adaptant automatiquement les optimisations utilisées en fonction de la forme et des caractéristiques optiques des objets étudiés.

Une telle approche doit permettre de limiter les temps de calcul grâce à :

- une importante baisse du nombre de rayons calculés inutilement. L'utilisation des projections pour calculer les différents types de rayon (primaires,secondaires et ombrage) permet de restreindre le domaine de représentation de l'objet donc de limiter le nombre de rayons lancés.

- une importante diminution du nombre d'objets testé pour chaque intersection. En effet, effectuer la décomposition d'un espace de travail en recherchant les subdivisions les plus appropriées au contexte de travail, au nombre, à la forme et la répartition des objets permet de construire des sous-espaces plus adaptés à l'application efficace d'un algorithme de rendu.

- une organisation des calculs liée aux caractéristiques des objets. Savoir rationaliser les calculs effectués permet définir des priorités aux traitements et d'apporter une efficacité différente, dans la mesure où elle n'est pas exclusivement calculatoire, aux calculs des intersections rayon-objet.

Chapitre 4

VALIDATION DE LA METHODE

Présentation des résultats

1. INTRODUCTION

Au-delà de l'intérêt théorique des méthodes d'optimisation, un point de passage obligé pour la validation de concepts reste le développement d'une application. Pour que les évaluations soient explicites nous avons voulu détacher les tests des calculs de toutes considérations matérielles. En effet, les temps de calcul sont trop dépendants des performances de la machine, de la qualité de programmation, de la manière dont sont représentés les objets. C'est pourquoi nous avons préféré exposer un bilan comparatif plutôt que de fournir une suite rébarbative de temps de calcul dont l'interprétation, trop souvent ambiguë, peut fausser les conclusions. Nous avons implanté une méthode test où les seules optimisations sont l'utilisation des englobants sur les objets. Tous les calculs de la décomposition adaptative seront comparés aux résultats obtenus par une méthode utilisant uniquement les englobants.

2. PRESENTATION DES CONDITIONS DE TEST

Pour que les comparaisons puissent être significatives, nous avons développé la décomposition adaptative à partir de l'algorithme utilisant les englobants. L'optimisation est en fait une couche supérieure faisant appel aux mêmes primitives de base que la méthode des englobants. Les procédures de calcul d'intersection rayons faces, d'affichage, ... sont identiques. Les scènes choisies présentent des caractéristiques plus liées à la mécanique qu'au design et les objets ne sont pas en situation dans un environnement complexe. Toutes les scènes ont été modélisées à partir de SACADO (Système Adaptatif de Conception et d'Aide au Développement par Ordinateur), développé au Laboratoire de Recherche en Informatique de Metz. Ces conditions permettent d'obtenir une intégrité sur les conditions de réalisation des tests et apportent plus de crédibilité à l'interprétation des résultats.

La version actuelle de SACADO est écrite en PASCAL sur une station de travail SUN Sparc2 GS utilisant le système d'exploitation UNIX. Les images sont affichées sur un écran 1280x1024 pixels et disposant de 24 plans de bits pour la couleur.

Un des soucis permanent de SACADO est de pouvoir être porté très facilement. C'est pourquoi, la partie logiciel graphique de base a été écrite en utilisant la norme X11. Cela garantit une indépendance totale par rapport au matériel.

Pour la partie "rendu", une interface utilisateur spécifique a été créée. "L'habillage" OpenLook est utilisé pour conserver le même aspect que l'ensemble des applications de l'environnement de développement OpenWindows. Cette interface permet le déplacement en temps réel de l'observateur dans la scène en représentation fil de fer, la description rapide et facile des sources lumineuses et des caractéristiques des objets. Cependant, X11 souffre d'un inconvénient majeur. S'il offre toutes les fonctionnalités nécessaires à la réalisation d'une application graphique, elles restent néanmoins de bas niveau. Cela ne facilite pas l'écriture d'une interface utilisateur qui soit à la fois performante et conviviale et qui profite au maximum du multifenêtrage. L'utilisation d'outils tels que les *widgets* (windows gadgets), qui, aujourd'hui, ne sont pas encore normalisés mais dont l'emploi tend tout de même à se généraliser, pallie cet inconvénient. Les boutons, ascenseurs et autres popupmenus permettent de concevoir des interfaces utilisateur très conviviales, de plus ils diminuent considérablement le temps de développement. Un exemple de fenêtres est présenté en annexe 3.

Pour la partie "rendu", une interface utilisateur spécifique a été créée. Les widgets OpenLook sont utilisés pour conserver le même aspect que l'ensemble des applications de l'environnement de développement OpenWindows. Cette interface permet le déplacement en temps réel de l'observateur dans la scène en représentation fil de fer, la description rapide et facile des sources lumineuses et des caractéristiques des objets. Une partie importante de cette interface est consacrée au paramétrage du tracé de rayons. Pour chaque type de rayon (primaire, secondaire et ombrage), l'utilisateur a le choix entre plusieurs types d'optimisations. De même pour la décomposition adaptative, il peut combiner les tests d'arrêt et modifier les valeurs limites, ce qui permet à l'utilisateur d'employer au mieux sa connaissance de la scène pour augmenter l'efficacité d'algorithme.

Grâce à une telle interface, l'affichage réaliste d'une scène n'est plus uniquement l'étape finale dans la création d'un objet, mais fait aussi partie intégrante de l'étape de conception. L'utilisateur peut vérifier visuellement et de manière réaliste la validité de son objet à chaque phase de la conception.

Pour juger plus précisément des performances et des limites de la méthode nous avons séparé les tests en plusieurs catégories. La première partie (le paragraphe 3) présente une série de tests s'appliquant à des scènes dont la géométrie des objets est plutôt favorable à la décomposition adaptative. L'ensemble des objets réfléchissants et transparents sont de type parallélépipédique. Nous avons créé des scènes types dont les fonctionnalités mécaniques sont plus que douteuses mais à l'intérieur desquelles ont retrouvé des types de faces très différentes comme les faces trouées ou les faces concaves.

Nous présenterons dans une seconde partie (paragraphe 4) un ensemble de résultats liés à des scènes moins favorables à l'application d'une optimisation par décomposition. Ces scènes servent aussi à montrer l'adaptabilité des algorithmes aux caractéristiques géométriques et optiques des objets, les performances baissant très sensiblement pour des scènes très défavorables.

Les principaux modèles utilisés pour valider la méthode sont présentés à l'annexe 1.

Tous le temps que nous donnons représentent des temps de calcul effectifs et non des temps CPU tout simplement parce que les différences de temps CPU n'étaient pas assez significatives pour des calculs d'images rapides.

3. PRESENTATION DE LA METHODE GENERALE

Nous donnons la signification de toutes les notations que nous utiliserons pour les différents tests.

nom	taille	optim	Ttail	Tnbob	Tvide	Ttaux	nbo	fen	taux	L	C
pré	traite	prim	pri E	sec	sec E	omb	omb E				

- nom** : est le nom du modèle (se référer à l'annexe 1)
- taille** : est la taille de l'image, 500 signifiera que l'image est de 500 par 500
- optim** : est l'optimisation choisie, E pour englobant et D pour décomposition adaptative
- Ttail** : prise en compte du test sur la taille minimale des sous-fenêtres
- Tnbob** : prise en compte du test sur le nombre d'objets minimum par sous-fenêtre
- Tvide** : prise en compte du test sur la présence de sous-fenêtres vides
- Ttaux** : prise en compte du test sur le taux d'occupation de la fenêtre
- nbo** : représente le nombre d'objet par sous-fenêtre
- fen** : représente la taille de la sous-fenêtre minimale en pixels (une taille de 100 représente une sous-fenêtre de 10 par 10 pixels)
- taux** : représente le taux d'occupation de la fenêtre étudiée
- L** : le calcul tient compte du lissage des objets de révolution
- C** : le calcul tient compte des faces concaves
- pré** : temps du prétraitement qui est essentiellement le temps des décompositions
- traite** : temps de calcul de l'image incluant le temps de prétraitement
- prim** : nombre de rayons primaires calculés
- prim E** : nombre d'intersections effectives pour les rayons primaires
- sec** : nombre de rayons secondaires calculés

sec E : nombre d'intersections effectives pour les rayons secondaires
omb : nombre de rayons d'ombrage calculés
omb E : nombre d'intersections effectives pour les rayons d'ombrage

Afin de simplifier la présentation de certains calculs les informations inutiles seront absentes des tableaux.

3.1. Optimisation des rayons primaires et d'ombrages

Nous présentons dans un premier temps un comparatif entre une optimisation par englobants et une optimisation par décomposition uniquement pour les rayons primaires et les rayons d'ombrage. Les scènes testées ne sont constituées que d'objets opaques. Plutôt que de présenter une liste interminable de chiffres nous avons préféré limiter le nombre d'exemples à trois mais choisir ces exemples de telle manière que nous retrouvions un large éventail de faces représentatives.

Le premier modèle présenté est celui des haltères. Cette scène a comme particularité de posséder des objets parallélépipédiques et des objets de révolution, ceux-ci présentant de grandes disparités au niveau de la taille des faces et certaines des faces des "poids" sont concaves.

Le second modèle possède des caractéristiques similaires si ce n'est que l'objet ne possède aucune face concave.

Le troisième modèle représentant un ensemble de sphères, est un exemple classique où tous les objets sont de révolution.

3.1.1. Les caractéristiques de la décomposition

Nous n'avons pas souhaité faire apparaître ici l'influence des tests de décomposition sur les temps de calcul. Nous reviendrons plus en détail sur l'influence des paramètres de décomposition au paragraphe 5.

Les caractéristiques retenues pour les tests d'arrêt sont :

nom	taille	opti	Ttail	Tnbo	Ttaux	Tvide	nbo	fen	taux	L	C
haltères	500	E								X	X
haltères	500	D	X	X	X	X	20	200	60	X	X
escalier sphères	500	E								X	
escalier sphères	500	D	X	X	X	X	20	200	60	X	

3.1.2. Les temps de calcul

Les différents temps de calcul sont présentés en heures, en minutes ou en secondes. Le temps de traitement incluent de temps de prétraitement

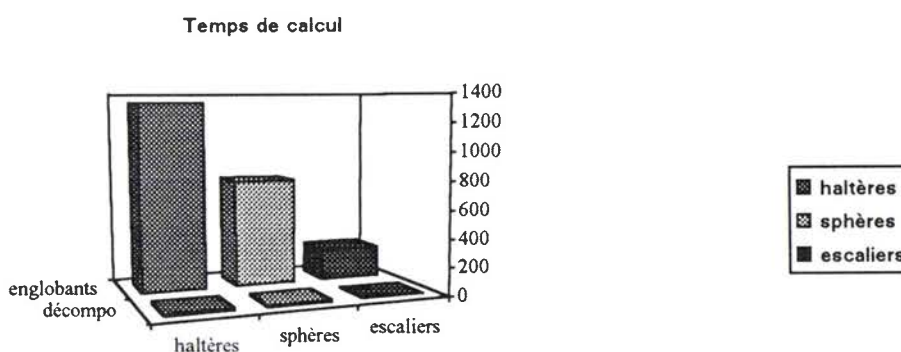
nom	taille	optim	pré	traite	prim	prim E	omb	omb E
haltères	500	E		22h 18	157700700	48120	19599728	6325
haltères	500	D	29"	17' 20"	74766	48120	50452	6325

escalier	500	E		3h 45'	73794254	97537	25105370	20010
escalier	500	D	8"	4' 06"	103274	97537	155810	20010

sphères	500	E		12h 50'	256752540	58025	75131625	261
sphères	500	D	36"	12' 30"	74861	58025	58675	261

3.1.3. Interprétation des résultats

Le graphique ci-dessous présente un histogramme des variations des temps de calcul (exprimés en minutes) pour chacun des trois modèles. La variation des valeurs entre la méthode des englobants et celle de décomposition est si importante que les graphiques des résultats de décomposition sont fortement aplatis. Nous aurions pu atténuer cette différence en prenant une échelle de valeurs logarithmique mais nous pensons qu'une importante disparité entre les colonnes symbolise mieux les différences exprimées par les calculs.



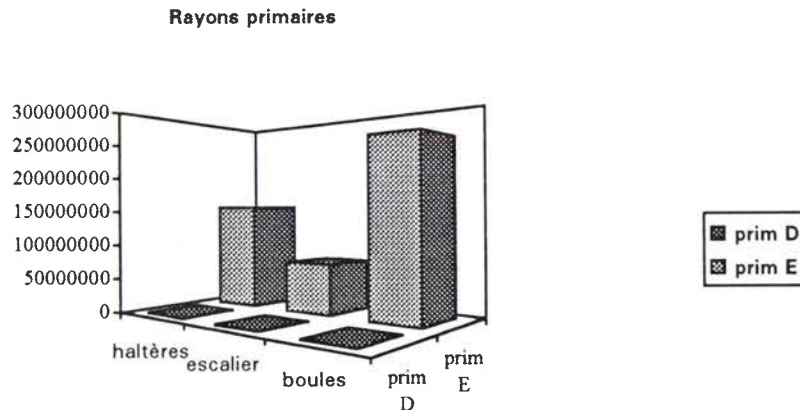
On constate que le gain de temps est de 98.7% pour le modèle des haltères et de 98.22% pour l'escalier et de 98.3% pour les sphères. Pour l'ensemble de scènes que nous avons testées, le gain de temps calculé uniquement sur les rayons primaires et d'ombrage n'est jamais descendu en deçà de 94%. Les gains minimaux ont été obtenus pour des scènes dont le taux de remplissage de l'image était proche des 100%, dans ces conditions, la décomposition adaptative ne peut isoler des zones vides d'information, le gain de temps est du uniquement à la diminution de la complexité dans chacune des sous-fenêtres, les performances ont donc naturellement tendance à baisser.

Une comparaison tout aussi importante est celle effectuée sur le nombre de rayons calculés. Si le nombre d'intersections effectives est identique pour les deux méthodes d'optimisation, le nombre de rayons calculés est par contre très différent.

La complexité de l'algorithme de base du tracé de rayons est en $O(F)$. Pour une image de $P \times P$ pixels contenant F faces, le nombre de calculs d'intersections est de $P \times P \times F$. Pour les modèles analysés, le nombre théorique des rayons primaires est de $2.7675 \cdot 10^8$ pour les sphères, de $0.735 \cdot 10^8$ pour les escaliers et de $1.75 \cdot 10^8$ pour les haltères.

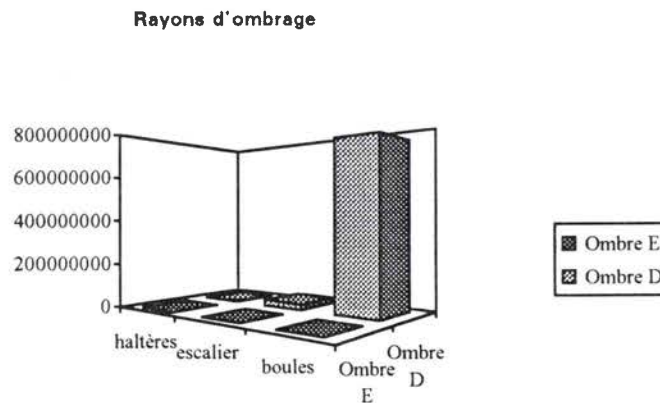
Si l'on exprime le nombre de rayons primaires calculés par rapport au nombre de rayons primaires effectifs, le taux de calcul de rayons "inutiles" est de 99.993% pour les sphères, de 99.87% pour les escaliers et de 99.97% pour les haltères. L'utilisation des englobants ne diminue pas fortement ces chiffres puisque nous obtenons des taux de 99.98% pour les sphères, de 99.85% pour les escaliers et de 99.96% pour les haltères.

Si l'on regarde maintenant l'histogramme présentant un comparatif entre les rayons effectifs primaires des deux méthodes, on remarque que le taux de rayons "inutiles" chute très fortement lors de l'utilisation de la décomposition adaptative.



Il devient égal à 46% pour les haltères, à 6% pour les escaliers et à 32% pour les sphères. Le fort pourcentage réalisé pour le calcul des escaliers est dû à deux choses. La première est que la scène ne comporte que des éléments dont la forme est bien approchée par le volume englobant parallépipédique. La seconde est que les objets sont relativement concentrés, ce qui permet d'avoir très peu de rayons ne coupant aucun objet. De manière générale on se situe plus aux alentours de 30 % de rayons calculés inutilement.

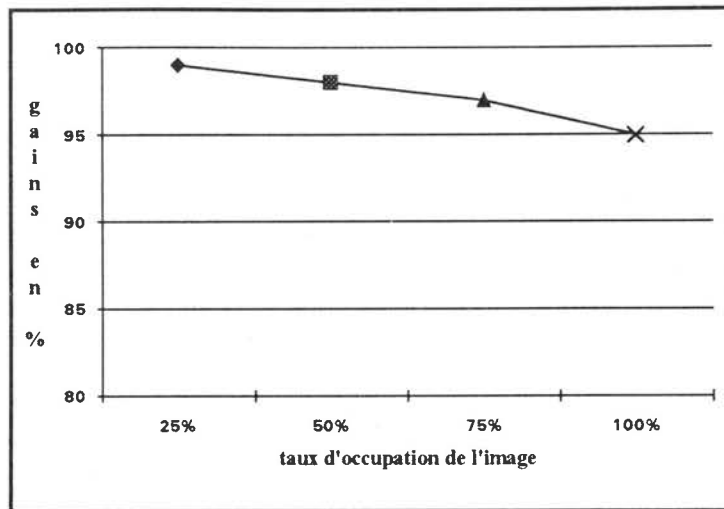
Bien que les sources lumineuses soient définies comme un observateur, nous ne retrouvons pas au niveau des rayons d'ombrage les mêmes rapports de valeurs que pour les rayons primaires. L'histogramme ci-dessous possède une configuration similaire à celui des rayons primaires mais les rapports de valeurs sont moins importants.



Le taux de rayons d'ombrage calculés inutilement avec la méthode des englobants est respectivement de 99.97% pour les haltères, de 99.99% pour les sphères et de 99.93% pour les escaliers. Le même calcul effectué avec la méthode de décomposition donne des résultats de 88% pour les haltères, de 99.6% pour les sphères et de 87% pour les escaliers. La très mauvaise performance des rayons d'ombrage des sphères est due au fait que la source lumineuse est placée pratiquement sur l'oeil, les objets étant éclairés plein champ, le nombre de rayons d'ombrage effectif devient très faible par rapport au nombre de rayons calculés (autant que de rayons primaires). Dans ces conditions, les résultats ne sont pas très significatifs, le gain moyen calculé sur l'ensemble des scènes traité est plus proche de 80%.

3.1.4. Conclusion

Les résultats de tests ont montré tout l'intérêt de l'application d'une décomposition adaptative aux rayons primaires et aux rayons d'ombrage. La méthode a permis de diminuer considérablement les temps de calcul en limitant l'application de l'algorithme à des zones spécifiques. Les temps de prétraitement ne sont aucunement pénalisant. En effet, pour l'image des haltères qui est une des images calculée le plus rapidement, le temps de décomposition ne représente que 3% du temps total, il n'est plus que de 1,2% pour la scène du moyeu et représente moins de 0.5% pour la scène de la bouteille. Au-delà des temps de calcul le point important est la forte diminution des rayons primaires inutiles et à un degré moindre, les rayons d'ombrage. La méthode idéale étant celle qui permettra d'obtenir un pourcentage de rayons non effectifs égal à zéro. Toutes les images sont référencées à l'annexe 2. Le graphique ci-dessous présente les gains de temps pour différents taux d'occupation de l'image.



3.2. Optimisation des rayons primaires, d'ombrages et réfléchis

Si la forme des objets n'influe pas le choix de l'optimisation des rayons primaires et d'ombrage, elle va déterminer le choix des options retenues pour les rayons réfléchis ou réfractés. On a vu au chapitre 3 que nous ne pouvions pas rechercher d'observateur symétrique si l'objet était de révolution. Si le cas se présente, nous changeons automatiquement la méthode d'optimisation pour analyser l'objet en ne considérant que les englobants. Ceci entraîne une dégradation des performances, nous proposons, à travers plusieurs exemples, d'évaluer l'influence de la géométrie des objets spéculaires sur les temps de calcul des images.

3.2.1. Les caractéristiques de la décomposition

Comme pour le paragraphe 3.1 nous limitons la présentation des résultats aux exemples caractéristiques permettant de montrer les avantages et les problèmes liés à la méthode. La première scène (escalier), la plus favorable, ne contiendra que des faces réfléchissantes planes. La seconde (bouteille) sera un panaché de faces réfléchissantes planes et de révolution. Enfin toujours la scène des sphères où tous les objets réfléchissants seront de révolution.

Les caractéristiques retenues pour les tests d'arrêt sont :

nom	taille	opti	Ttail	Tnbo	Ttaux	Tvide	nbo	fen	taux	L
escalier	500	E								X
escalier	500	D	X	X	X	X	20	200	60	X
bouteille	500	E								X
bouteille	500	D	X	X	X	X	10	400	60	X
sphères	500	E								X
sphères	500	D	X	X	X	X	10	400	60	X

3.2.2. Les temps de calcul

Les tableaux ci-dessous présentent les temps de calcul des différentes images référencées en annexe 2.

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
escalier	500	E		5h 06'	74977350	77777	20861800	6075	22950150	10825
escalier	500	D	14"	8' 30"	92326	77777	70724	6075	83850	10825

Temps de calcul quand 33% des sphères sont réfléchissantes

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
sphères	500	E		21h 15'	330512400	124374	53987472	6899	168376150	2224
sphères	500	D	17"	3h 30'	158651	124374	53987472	6899	131273	2224

Temps de calcul quand 100% des sphères sont réfléchissantes

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
sphères	500	E		26h 20'	330512400	124374	161188704	23348	186923125	4897
sphères	500	D	17"	10h 12'	158651	124374	161188704	23348	147722	4897

Temps de calcul de la scène bouteille avec uniquement le plateau réfléchissant

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
bouteille	500	E		34h	546518575	81043	105649900	18588	181317250	117786
bouteille	500	D		3h 06'	124951	81043	48572322	18588	99631	11786

Temps de calcul de la scène bouteille avec plateau et verres réfléchissants

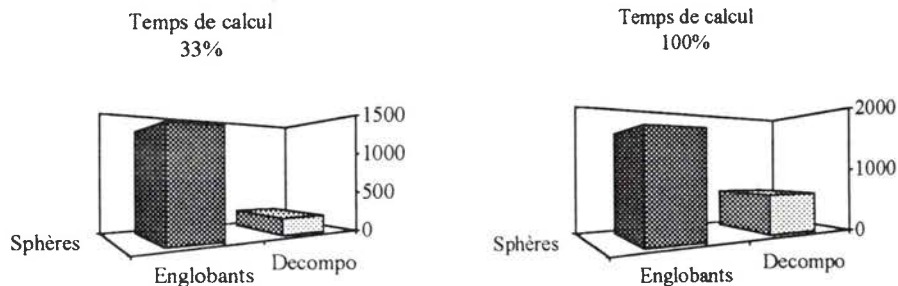
nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
bouteille	500	E		36h 55'	546518575	81043	143795300	24180	192032375	1380
bouteille	500	D	35"	5h 10'	124951	81403	85526214	24180	165223	1380

3.2.3. Interprétation des résultats

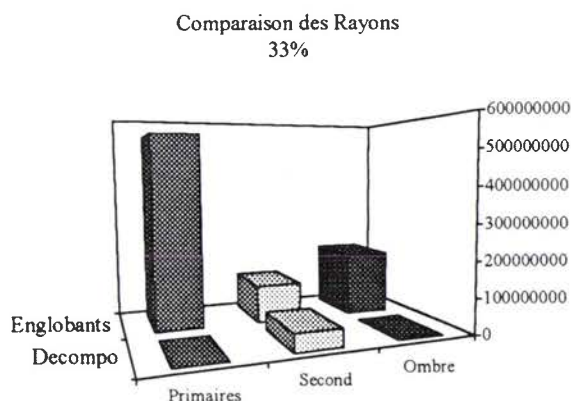
3.2.3.1. Scènes ne contenant que des objets réfléchissants de révolution

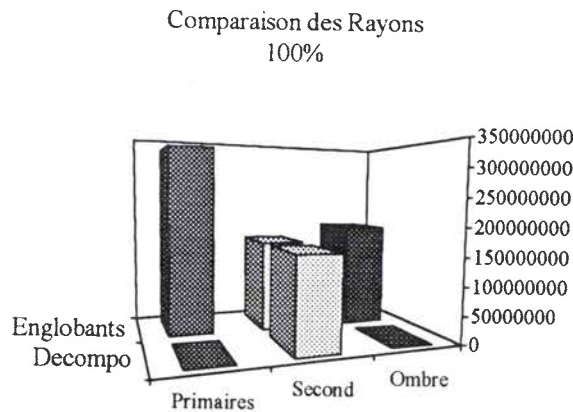
Le cas le moins favorable à l'application d'une décomposition adaptative reste une scène ne contenant que des objets réfléchissants de révolution. Nous nuancerons cette affirmation en précisant que seuls les calculs liés aux rayons secondaires diminuent les temps de calcul, les optimisations appliquées aux rayons primaires et, à un degré moindre, aux rayons d'ombrage permettant toujours de diminuer les coûts de traitement dans les proportions définies au paragraphe 3.1.

La visualisation des temps de calcul pour une scène dont 33% des objets sont spéculaires montre une différence sensible entre les temps des englobants et les temps des décompositions. Avec un tiers d'objets réfléchissant les gains chutent en moyenne aux alentours de 83%, ils ne représentent plus que 62% pour une scène où tous les objets sont de révolution et réfléchissants.



Cette différence est encore plus marquée si l'on compare les histogrammes des différents types de rayons.





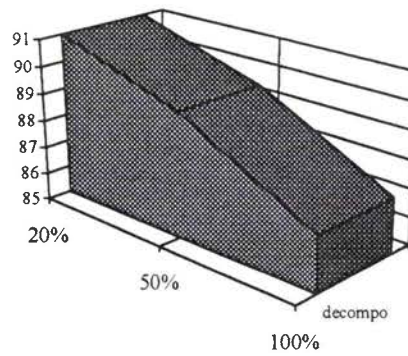
On remarque que la dégradation des temps de calcul est essentiellement due au fait que le nombre de rayons secondaires calculés est identique pour les deux méthodes d'optimisation. La décomposition adaptative n'est plus appliquée que sur les rayons primaires et d'ombrage.

3.2.3.2. Scènes contenant des objets réfléchissants de type plan et des objets de révolution

La deuxième série de tests a été effectuée sur des scènes plus représentatives d'une certaine réalité. En effet, il est courant de trouver au sein d'une même image des objets spéculaires de révolution et des objets spéculaires de type plan. Pour des scènes où cohabitent les deux formes de surfaces réfléchissantes, il est possible de réaliser une application partielle et sélective de la décomposition adaptative. Une analyse du contenu singularise l'optimisation appliquée à chacun des objets et particularise les choix effectués pour chacune des faces. Afin de valider la méthode de décomposition adaptative sur de telles sortes de scènes nous avons défini le modèle de la bouteille où l'on trouve des objets strictement de révolution comme les verres ou la bouteille et des objets mixtes comme le plateau présentant la particularité d'avoir des grandes faces planes réfléchissantes (le fond) et de petites faces de révolution (les bords).

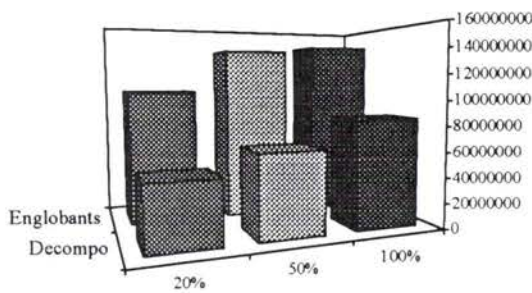
Comme pour la scène des sphères nous avons graduellement augmenté le nombre d'objets réfléchissants de révolution. La première série de calculs présentés au paragraphe 3.2.2 correspond à la scène de la bouteille où seul le plateau est réfléchissant. Une représentation des temps de calcul montre la dégradation des performances en fonction du pourcentage d'objets spéculaires de révolution pris en compte pour la scène de la bouteille. Si le calcul ne prend en compte que 20% d'objets spéculaires (uniquement le plateau), les gains de temps par rapport à la méthode des englobants est de l'ordre de 91%. Ce gain avoisine les 87% si l'on considère comme réfléchissant tous les objets de révolution de la scène.

Gains suivant le pourcentage d'objet de révolution

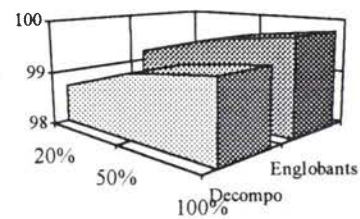


Cette diminution des performances corrobore la tendance exprimée lors de l'analyse de scènes ne contenant que des objets de révolution. Cependant, la présence d'importantes faces planes réfléchissantes, donc la possibilité d'appliquer une décomposition adaptative, permet de conserver en moyenne des gains proches de 90%. Cette dégradation des performances se retrouve au niveau des rayons secondaires et d'ombrage calculés inutilement. Les histogrammes ci-dessous présentent cette variation en fonction du pourcentage d'objets réfléchissant de la scène.

Comparaison des Rayons secondaires

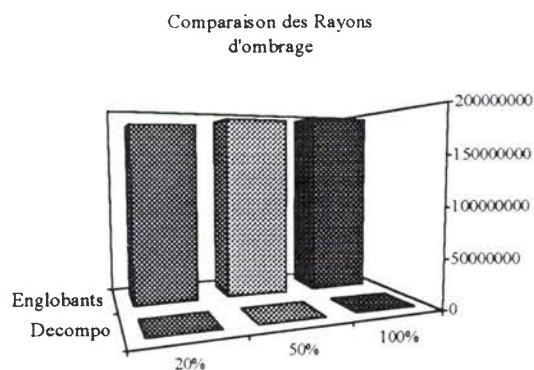


Pourcentage de rayons secondaires calculés inutilement



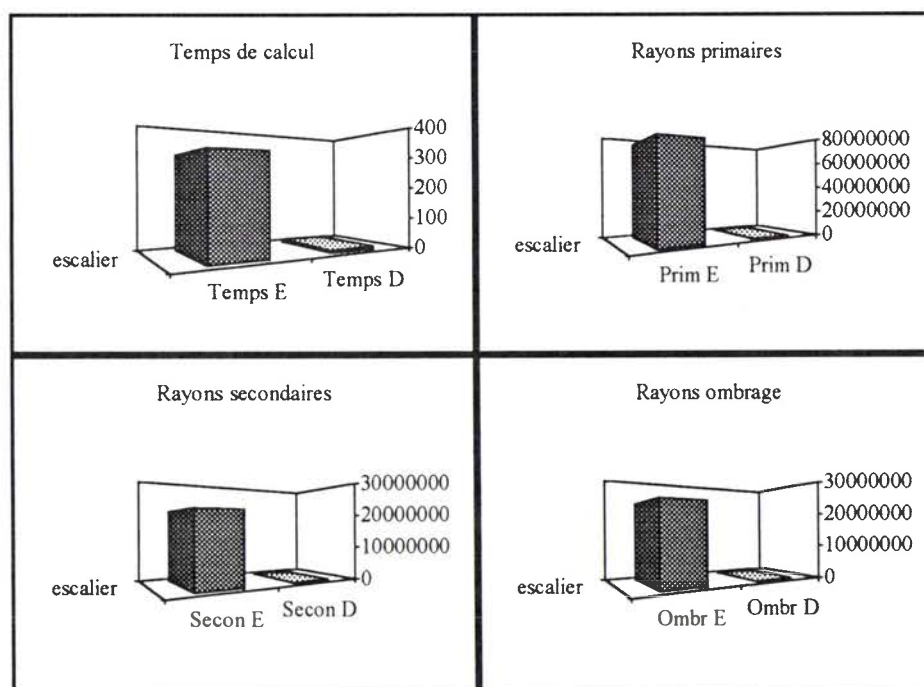
La baisse de performance est nettement moins sensible au niveau des rayons d'ombrage.

Le rapport entre le nombre de rayons calculés par la méthode des englobants et par la méthode de décomposition est toujours aussi important. Nous n'effectuons ici aucune analyse sur le nombre de rayons d'ombrage calculés inutilement par les deux méthodes, cette valeur est peu significative, le nombre d'intersections effectives étant trop dépendant de la position des sources lumineuses.



3.2.3.3. Scènes ne contenant que des objets réfléchissants de type plan

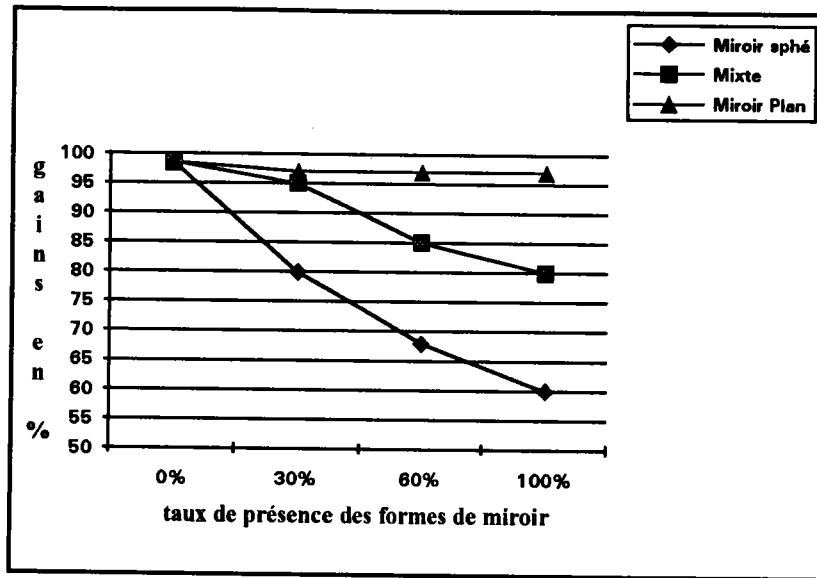
Le cas le plus favorable à l'utilisation d'une optimisation par décomposition est une scène où toutes les faces spéculaires sont planes. On retrouve des différences identiques aussi bien au niveau des temps de calcul qu'au niveau des rayons primaires, des rayons d'ombrage mais aussi des rayons secondaires réfractés. Les histogrammes ci-dessous visualisent pour les quatre paramètres les rapports de valeurs pour la scène de l'escalier.



L'information importante que l'on doit ressortir de l'analyse des histogrammes est que l'on a, grandeur d'échelle mis à part, une variation pratiquement identique des rayons primaires, d'ombrage et secondaires. Cet écart entre les méthodes est de 98.8% pour les rayons primaires, de 97% pour les rayons secondaires et les rayons d'ombrage. Cette variation quasiment linéaire fait que les performances, au niveau des temps de calcul, ne sont pas dégradées si les faces réfléchissantes appartiennent à des objets de type plan. Le gain moyen pour l'ensemble des scènes testées ne contenant que des objets opaques est de l'ordre de 98%. Nous retrouvons les mêmes performances pour des scènes ne contenant que des miroirs plans puisque les gains avoisinent les 97% toujours par rapport à la méthode des englobants. Il semble utile de préciser que les temps de décomposition varient suivant les mêmes proportions et que le coût global de prétraitement n'a jamais excédé 4% du temps total, pour les images les calculs les plus rapides (quelques minutes) et reste nettement inférieur à 1% pour des images nécessitant des temps de calculs de l'ordre de l'heure.

3.2.4. Conclusion

L'application de la décomposition adaptative aux scènes contenant des objets spéculaires a montré que les performances étaient fortement liées à la géométrie des objets. Pour des scènes ne contenant que des objets de révolution réfléchissants, l'utilisation des englobants comme unique optimisation des rayons secondaires pénalise fortement les temps de calcul. L'idéal serait d'utiliser une méthode d'optimisation plus adaptée au traitement des rayons secondaires comme les octree. Néanmoins, pour l'ensemble des scènes que nous avons analysées, le gain le plus faible, pour des scènes entièrement composées d'objets de révolution, approche les 60%. Par contre si la scène contient des surfaces spéculaires de type plan et de révolution, on mélange pour les rayons secondaires des optimisations par décomposition et des optimisations utilisant des englobants, dans ces conditions, les gains avoisinent les 80%. On retrouve le même niveau de performance identique aux rayons primaires et d'ombrage, pour les rayons secondaires, si la scène ne contient que des miroirs de type plan. Dans ce cas la courbe des gains est pratiquement linéaire et la dégradation des performances liées à l'utilisation d'une décomposition adaptative est pratiquement nulle. Le graphique ci-dessous récapitule les gains de temps pour les différents types d'objets réfléchissants en faisant varier la proportion d'objets spéculaires à l'intérieur d'une même scène.



3.3. Optimisation des rayons primaires, d'ombrage et réfractés

Comme nous avons analysé l'influence de la géométrie des objets spéculaires sur le temps de calcul, nous nous proposons de réaliser la même démarche pour les objets transparents. Nous avons défini les limites de l'application de la décomposition adaptative aux objets transparents au chapitre 3. Comme pour les objets spéculaires nous avons choisi un nombre limité d'exemples de scènes couvrant le plus large éventail possible de situations. Pour cela nous présentons les calculs réalisés sur deux scènes type: la poulie et les haltères (voir annexe 1).

3.3.1. Les caractéristiques de la décomposition

Comme pour le paragraphe 3.1 nous limitons la présentation des résultats aux exemples caractéristiques permettant de montrer les avantages et les problèmes liés à la méthode. Nous avons choisi, pour illustrer nos propos, le modèle des haltères et celui de la poulie.

nom	taille	opti	Ttail	Tnbo	Ttaux	Tvide	nbo	fen	taux	L	C
haltères	500	E								X	X
haltères	500	D	X	X	X	X	10	100	60	X	X
poulie	500	E								X	X
poulie	500	D	X	X	X	X	10	400	60	X	X

3.3.2. les temps de calcul

Nous avons choisi la scène des haltères parce qu'elle permet de représenter les cas extrêmes à l'application de la méthode de décomposition. La surface parallélépipédique symbolise une vitre à travers laquelle nous apercevons un ensemble d'objets. Nous analyserons cette scène pour un coefficient de transparence différent de 1 et pour un coefficient égal à 1.

Le premier point correspond à une situation où une scène entièrement vue à travers une vitre est déformée par la diffraction de la lumière dans la matière. Dans ces conditions, on ne peut utiliser la décomposition adaptative pour calculer les rayons réfractés, elle ne pourra s'appliquer qu'aux rayons primaires et d'ombrage et uniquement pour l'objet transparent.

La seconde image calculée sera toujours la scène des haltères vue à travers une vitre mais cette fois, le coefficient de réfraction sera égal à 1. L'image ne subit aucune déformation due à la réfraction, nous pouvons utiliser une optimisation par décomposition.

La deuxième scène analysée sera celle de la poulie pour laquelle plusieurs pièces seront transparentes. Nous avons choisi ce modèle pour représenter des images où des objets seront partiellement vus à travers des surfaces transparentes.

Le coefficient de transparence est différent de 1

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
haltères	500	E		90h 53'	178517500	158000	223370000	165275	225193150	165275
haltères	500	D	45"	31h 28'	160000	158000	223370000	165275	225193150	165275

Le coefficient de transparence est égal à 1

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
haltères	500	E		92h	178517500	170973	223370000	175758	225193150	196502
haltères	500	D	1' 7"	53'	191565	170973	341946	175758	346731	196502

Le coefficient de transparence est différent de 1 (0.99)

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
poulie	500	E		300 h	28790700	177561	408261600	295144	1025132500	397124
poulie	500	D	1' 54"	85 h	251001	177561	223370000	295144	996211	397124

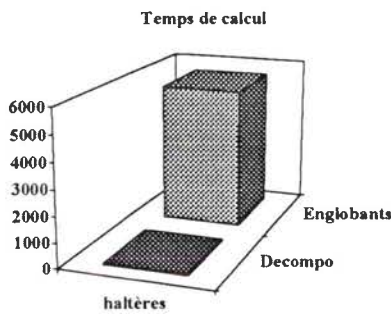
Le coefficient de transparence est égal à 1

nom	taille	optim	pré	traite	prim	prim E	sec	sec E	omb	omb E
poulie	500	E		300 h	28790700	117561	408261600	295144	102132500	397124
poulie	500	D	26"	78 h	251001	177561	220754196	295144	945410	397124

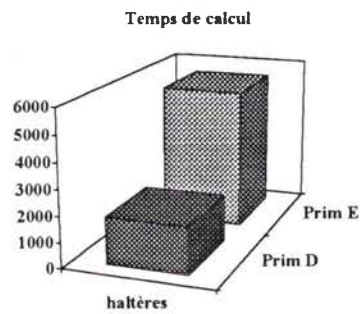
3.3.3. Interprétation des résultats

Analyse des valeurs

Lorsque les caractéristiques des objets permettent la mise en oeuvre de la méthode de décomposition, on retrouve un peu les mêmes niveaux de performances, que ce soit pour les rayons primaires, réfléchis ou réfractés. Les différents graphiques présentés ci-dessous confirment la tendance que l'on dégageait de l'analyse des rayons primaires et réfractés lorsque le coefficient de réfraction est égal à 1. On remarque par contre que la différence des temps de calcul entre une optimisation par décomposition et une optimisation par englobant se réduit pour retrouver des gains proches de 60%.

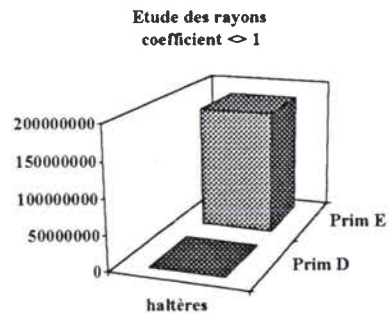
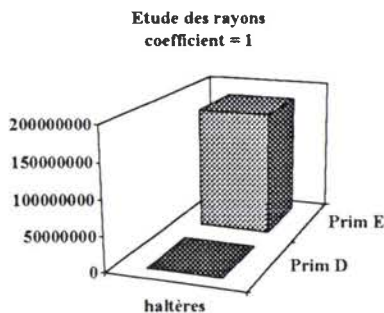


coefficient = 1

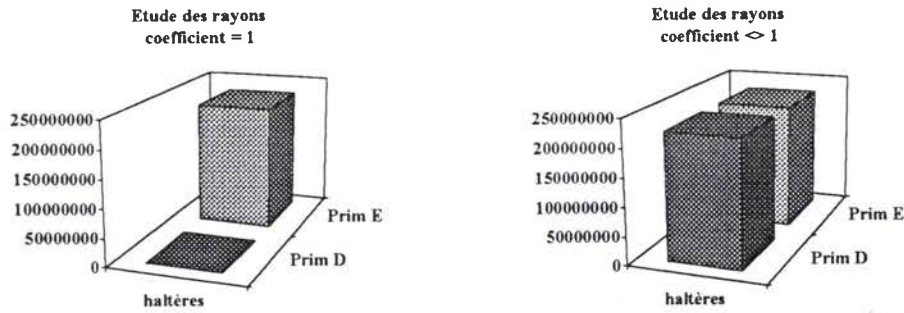


coefficient < 1

On retrouve les mêmes tendances si on analyse le comparatif entre les rayons primaires calculés et les intersections effectives.

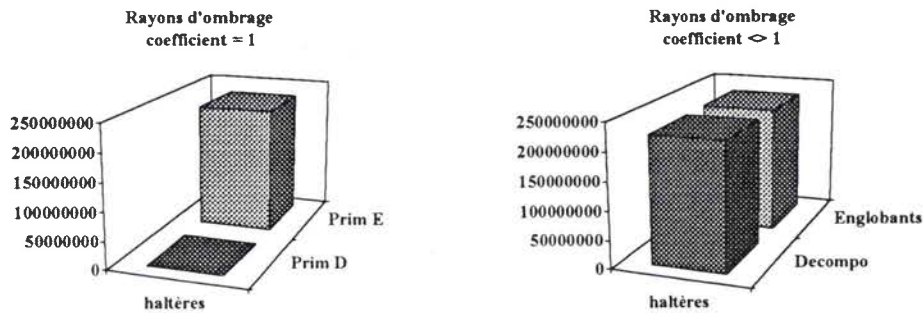


Par contre, le comparatif des rayons secondaires présente une toute autre configuration.



La différence au niveau des temps de calcul s'explique essentiellement par la différence du nombre de rayons secondaires calculés. Lorsque le coefficient de l'objet transparent est différent de un, on ne peut plus appliquer une optimisation par décomposition adaptative, nous sommes obligés de calculer les rayons réfractés en utilisant uniquement les englobants d'objets.

Nous retrouvons la même forme de graphique pour les rayons d'ombrage.



3.3.4. Conclusion

Nous avons choisi deux exemples significatifs permettant de mettre en évidence l'importante variation possible des gains en fonction des caractéristiques optiques des matériaux. Temps de calcul à l'appui, on remarque que les objets transparents restent ce qu'il y a de plus pénalisant pour la méthode. Ceci s'explique par la nécessité de calculer le rayon qui traverse la matière. Néanmoins, les gains sont fortement dépendants du coefficient de réfraction des objets transparents. Dans certains cas les rayons traversent la matière sans subir de déviation nous pouvons appliquer la méthode de décomposition adaptative. Dans ce cas on retrouve des niveaux de performance supérieurs à 95%. Par contre si le coefficient de réfraction est différent de un, les gains peuvent, en fonction du pourcentage d'objets transparents de la scène, chuter jusqu'à des valeurs avoisinant les 65%.

4. INFLUENCE DES TESTS D'ARRÊT

Dans l'optique d'une adaptation automatique des tests et de leur valeur à la scène, nous avons étudié l'influence des tests sur le temps de calcul afin de trouver le meilleur compromis possible en fonction du nombre d'objets et de leur répartition. Une première partie de l'étude a consisté à trouver les meilleures valeurs pour les tests de "*fenêtre minimum*", "*nb objets*", "*taux d'occupation*". Une seconde partie a consisté à trouver la meilleure organisation des tests d'arrêt en fonction du contenu de la scène. Nous proposons ensuite une organisation et des valeurs pour les tests d'arrêt en fonction d'un type de scène. Les choix proposés ne sont que la conclusion d'une suite de tests effectués sur des scènes types, ils seront donc sujet à débat.

4.1. Recherche des valeurs à attribuer aux paramètres

Afin de trouver le meilleur compromis pour les valeurs des tests d'arrêt, nous avons choisi des scènes différentes de part l'organisation et la taille des faces des objets. Pour chacune de ces scènes nous avons fait varier un à un les différents paramètres et nous avons relevé les temps de calcul pour chacune des images.

4.1.1. Variation du nombre d'objets par sous-fenêtre

Nous avons analysé dans un premier temps l'influence du nombre d'objets maximum dans une sous-fenêtre de décomposition. Nous avons effectué une analyse sur l'ensemble des scènes mais nous ne présentons les résultats chiffrés que de deux exemples significatifs, à savoir l'escalier et les sphères. La première des deux scènes est constituée à la fois de grandes faces planes et de petites faces de révolution, la scène présente à la fois une grande concentration d'objets au niveau de l'axe de l'escalier et des zones ne contenant que très peu d'objets. La seconde scène a comme particularité de présenter des zones bien séparées dans lesquelles on trouve des concentrations importantes de faces toujours de petites dimensions.

Pour effectuer les relevés de temps de calcul, nous avons fixé les autres paramètres à savoir la taille de la fenêtre minimale et le taux d'occupation.

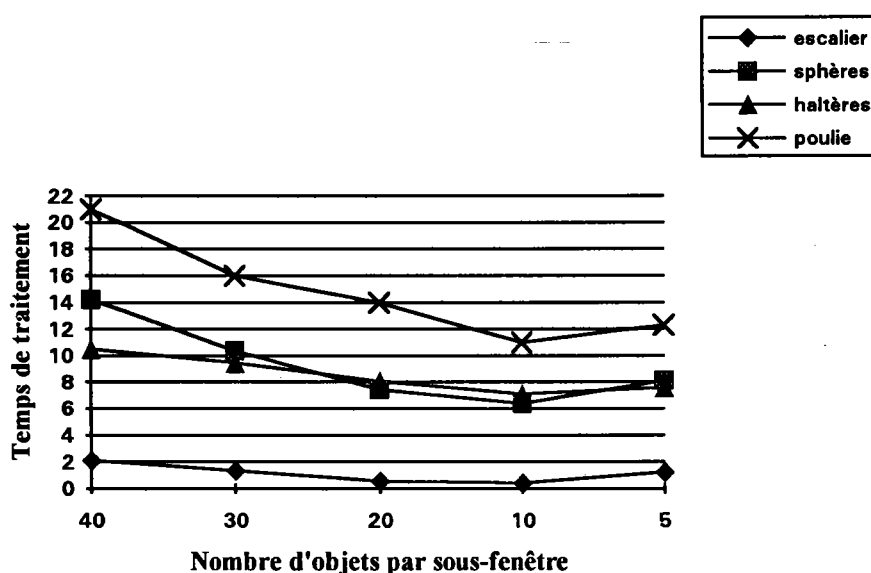
Escalier: 284 faces , tous les objets sont opaques, taille de l'image 200X200

Nb Objets	Fenêtre min	Taux	Prétraitement	Traitement
40	200	60	5"	2'12"
30	200	60	7"	1'32"
20	200	60	12"	58"
10	200	60	15"	43"
5	200	60	21"	1'23"

sphères: 1296 faces tous les objets sont opaques, taille de l'image 200X200

Nb Objets	Fenêtre min	Taux	Prétraitement	Traitement
40	200	60	36'''	14'21"
30	200	60	45'''	10'38"
20	200	60	1'05'''	7'45"
10	200	60	1'37"	6'38"
5	200	60	1'51"	8'10"

Afin de réaliser une analyse comparative plus explicite nous regroupons sur un même graphique les courbes des temps de calcul trouvées pour l'ensemble des scènes testées.



Une analyse des résultats fait ressortir une valeur pour laquelle les temps de calcul présentent un extremum. En deçà d'un nombre d'objets se situant aux environs de 10, la courbe des calculs a tendance à s'infléchir et à remonter. Cette augmentation s'explique par les temps d'évaluation d'un arbre de décomposition plus important. Ce n'est aucunement un accroissement du temps de décomposition qui pénalise le plus les résultats, celui n'excède pas quelques secondes. Le ralentissement des calculs est dû uniquement à l'évaluation accrue de l'arbre de décomposition.

4.1.2. Variation de la valeur du taux d'occupation

Il est relativement difficile d'évaluer seule l'influence de la variation du taux d'occupation sur les temps de calcul. En effet ce paramètre n'est jamais utilisé comme test d'arrêt unique, il se combine obligatoirement avec la taille de la sous-fenêtre ou le nombre d'objets.

La valeur affectée au taux d'occupation va varier en fonction du type de scène analysée. Nous n'avons pas trouvé une valeur, comme pour le test *nombre d'objets* ou le test *taille minimale* de la fenêtre, qui puisse convenir pour tous les types de scène. Si les objets de la scène ont tendance à être concentrés sur une partie de l'image, donner une valeur élevée au taux d'occupation, associé au paramètre *nombre d'objets* va permettre de limiter rapidement la décomposition. Par contre si les objets sont clairsemés, attribuer une valeur faible au taux d'occupation va faire apparaître plus facilement des zones vides d'information. C'est pourquoi nous avons utilisé un taux d'occupation de l'ordre de 70% pour des images comme la poulie, les escaliers, taux que nous avons ramené à 40% pour des scènes présentant une séparation nette des objets comme les sphères ou même certaines vues du moyeu.

4.1.3. Variation de la taille de la sous-fenêtre

La taille de la sous-fenêtre est certainement le paramètre le plus influent pour la décomposition adaptative. Si les deux tests précédents, à savoir le taux d'occupation et le nombre d'objets minimum, n'étaient que des tests d'orientation, celui de la taille minimale de la subdivision est en plus un test d'arrêt. Il se situe, dans l'analyse, au même niveau que *fenêtre vide et élément unique*. La constitution d'une maquette [GAR 91] avait permis de déterminer une sous fenêtre en deçà de laquelle les performances se détérioraient. Cette première étude, appliquée uniquement aux rayons primaires et d'ombrage, faisait ressortir que le meilleur compromis de taille pour une subdivision minimale était le 10000^{ème} d'une fenêtre originale de 1000 par 1000 pixels soit une sous-fenêtre de 100 pixels. Nous avons souhaité confirmer ces résultats en analysant l'influence de la taille de cette sous-fenêtre non seulement sur les rayons primaires et d'ombrage mais aussi sur les rayons secondaires. Pour cela nous avons choisi un jeu de tests où les scènes présentaient obligatoirement des objets spéculaires ou transparents. Afin de mieux juger de l'influence du test nous nous sommes placés dans des conditions favorables à la décomposition adaptative, c'est à dire que toutes les faces spéculaires sont planes et les faces transparentes ont un coefficient de réfraction égal à un. Nous ne présentons que deux résultats chiffrés correspondants à la scène de l'escalier, nous ferons apparaître uniquement sur le graphique les résultats de l'analyse des autres scènes.

Escalier: 284 faces , le sol est réfléchissant, taille de l'image 500X500

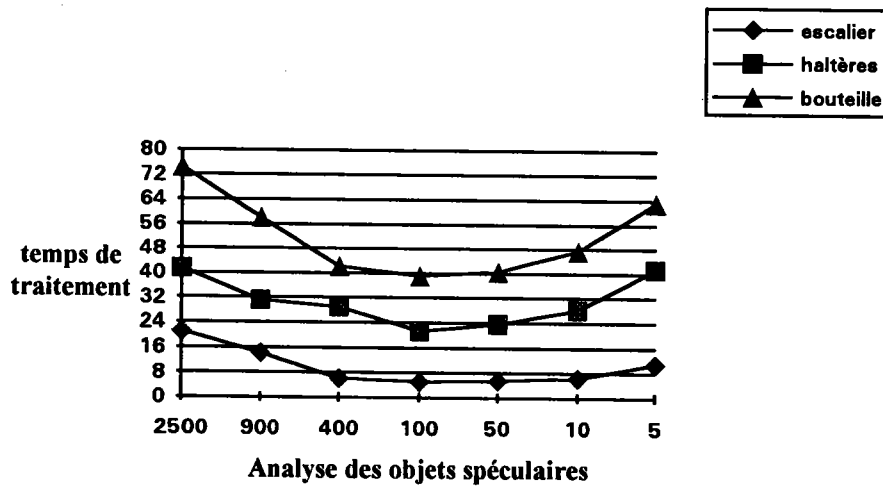
Nb Objets	Fenêtre min	Taux	Prétraitement	Traitement
10	2500 (50x50)	60	3"	21'12"
10	900 (30x30)	60	7"	14'23"
10	400 (20x20)	60	12"	7'20"

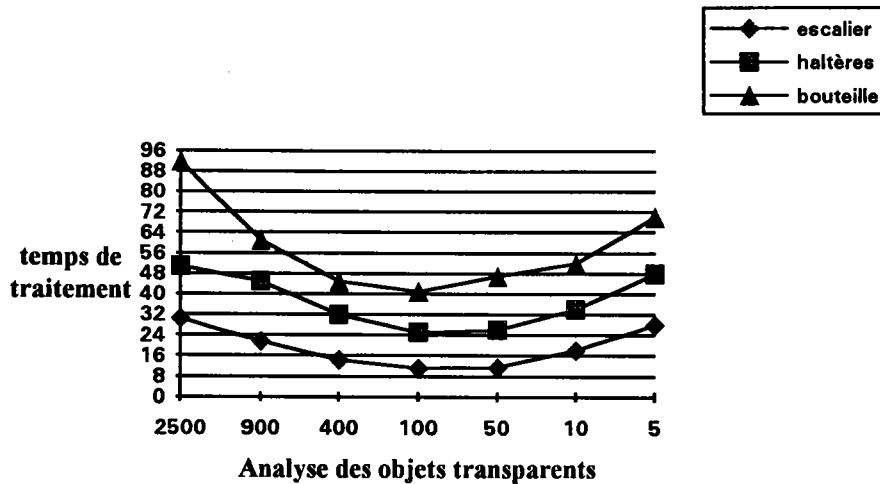
10	100 (10x10)	60	23"	5' 06"
10	50 (10x5)	60	33"	5'54"
10	10 (5x2)	60	41"	6'35"
10	5 (5x1)	60	1'10"	11'

Escalier: 284 faces , le sol est transparent, taille de l'image 500X500

Nb Objets	Fenêtre min	Taux	Prétraitement	Traitement
10	2500 (50x50)	60	7"	41'12"
10	900 (30x30)	60	11"	28'13"
10	400 (20x20)	60	19"	21'28"
10	100 (10x10)	60	27"	11' 21"
10	50 (10x5)	60	39"	12'34"
10	10 (5x2)	60	51"	14'55"
10	5 (5x1)	60	1'30"	18'35"

Nous présentons l'ensemble des résultats sur les graphiques ci-dessous.





L'allure générale des deux séries de courbes confirme les résultats trouvés lors de la réalisation de la première maquette. On trouve une valeur de seuil se situant dans un intervalle allant de 50 à 100. En deçà d'une sous-fenêtre de cette valeur, la méthode de décomposition adaptative est fortement pénalisée. Les temps de calcul augmentent fortement si l'on décompose jusqu'à obtenir des sous-fenêtres proches du pixel. En effet, l'arbre de décomposition devenant plus profond et les sous-fenêtres plus petites, à la fin de l'étude d'une subdivision on devra à nouveau parcourir l'arborescence afin de trouver la plus petite sous-fenêtre contenant le prochain point d'intersection. Une recherche sur un arbre quaternaire a une complexité qui est $O(H)$ où H est la hauteur de l'arbre, répéter cette recherche pour les rayons primaires, d'ombrage, secondaires et d'ombrage secondaire diminue les performances globales de la méthode.

4.1.4. Conclusion

Une analyse effectuée sur les différents paramètres de la décomposition a permis de trouver un ensemble de valeurs pour lesquelles les performances semblent optimum. Il est évident que ces valeurs sont spécifiques à l'application que nous avons développée et ne peuvent être généralisées sans précaution à toutes les méthodes utilisant des décompositions. Cependant on a montré, chiffres à l'appui, qu'un échantillonnage excessif peut introduire une explosion des temps de calcul et que la meilleure sous-fenêtre se situe entre 50 et 100 pixels. On a montré qu'il était plus intéressant de traiter "intelligemment" des sous-fenêtres contenant un nombre d'objets proche de 10 plutôt que de décomposer abusivement sans tenir compte du contenu des subdivisions.

4.2. Choix des tests d'arrêt

Toujours dans un souci d'améliorer des performances de calcul nous nous sommes interrogés sur la nécessité d'avoir en permanence tous les tests liés à la décomposition ? Est-il possible de dégager, des scènes analysées, un certain nombre de caractéristiques permettant de limiter le nombre de tests mis en oeuvre ?

Si nous avons défini ces tests dans un cadre global il est évident que nous pouvons trouver des scènes très particulières pour lesquelles nous avons la capacité de limiter la prise en compte de certains tests. L'idéal serait qu'une analyse détermine automatiquement quels seront les tests nécessaires à une décomposition optimale. Dans la version actuelle de la décomposition adaptative le choix est entièrement laissé à l'initiative de l'opérateur, il a la possibilité de modifier les tests mis en oeuvre en sélectionnant simplement le champ test à l'écran. Nous ne prétendons pas donner ici une liste exhaustive des scènes pour lesquelles nous pouvons restreindre les tests mis en oeuvre, ceci sera l'objet d'une étude ultérieure. Nous essayons seulement de mettre en évidence des situations particulières pour lesquelles une rapide analyse peut encore améliorer les traitements.

4.2.1. Scènes occupant toute la fenêtre

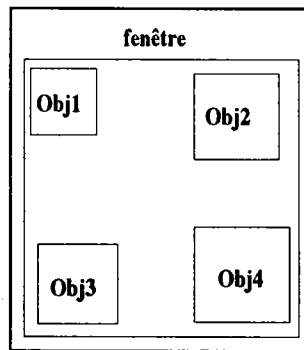
Il est courant de trouver, dans les images de synthèse dédiées à la conception artistique plus qu'en CAO mécanique, des scènes ne présentant aucune zone vide d'information. Pour ce genre de scène le test de *fenêtre vide* perd tout son sens et devient obsolète. Il devient alors inutile de le considérer lors de la décomposition. Il semble évident que la non prise en compte d'un tel test n'aura pas, dans ce cas, une répercussion importante sur le temps global de calcul. En effet, ne pas considérer le test de *fenêtre vide* peut éventuellement diminuer les temps de prétraitement mais ceux-ci ne représentent souvent qu'une infime partie du temps total de calcul d'image. L'intérêt est d'affiner une analyse en exploitant au mieux des connaissances issues d'une analyse du contenu.

4.2.2. Scènes ne contenant qu'un objet.

C'est un exemple de scènes particulières où seul le test *objet unique* est à considérer. Les tests *Test Taux*, *fenêtre min* et *nb objets* ne sont plus significatifs. Là encore, l'influence sur les temps de calcul est à démontrer, cependant il semble que les optimisations à venir devront obligatoirement passer par une meilleure compréhension du contenu. Au-delà d'une simple optimisation de l'utilisation de tests, il faut comprendre qu'une analyse adaptative permet de rationaliser les traitements même si ceux que nous avons décrit peuvent sembler parfois simplistes.

4.2.3. Scènes où les objets sont séparés

La dernière forme d'organisation des objets permettant de sélectionner plus particulièrement certains tests est une scène où les objets sont séparés. La figure ci-dessous présente une scène composée de quatre objets n'ayant aucune intersection entre eux.



L'utilisation du test de taux d'occupation devient inutile à condition de choisir pour le test *nb Objets* une valeur inférieure à 4. Une décomposition de la scène isolera les quatre objets et permettra de simplifier le traitement à l'intérieur de chacune des sous-fenêtres. Là encore, l'exemple choisi peut sembler simpliste mais il est plus important de retenir la démarche plus que le résultat.

4.2.4. Conclusion

Bien que nous n'ayons pas effectué une étude chiffrée de l'influence d'une prise en compte partielle des tests d'arrêt, nous pensons qu'il est important d'adapter les tests liés à la décomposition. Nous avons dégagé une liste non exhaustive de situations pour lesquelles tous les tests n'étaient pas obligatoires. Etre en mesure de trouver automatiquement, pour une scène quelconque, le meilleur compromis au niveau des tests d'arrêt reste une idée très intéressante qui peut sans doute être étendue à des notions plus générales comme par exemple la déduction automatique des optimisations à mettre en oeuvre. On peut imaginer un système capable de déduire automatiquement, en fonction d'une analyse du contenu, quels sont les formes d'optimisation les plus adaptées à la situation rencontrée.

5. CONCLUSION

Les résultats obtenus ont montré tout l'intérêt d'une décomposition souple capable de s'adapter au contenu de la scène analysée et aux compétences de l'opérateur. La séparation entre le traitement de la décomposition et la gestion des tests d'arrêt permet de personnaliser le prétraitement à chacune des scènes donc d'optimiser les traitements.

Il semble important de faire ressortir deux points forts de la méthode:

- La décomposition, qui peut être appliquée comme une tâche peut être considérée comme un modèle pour l'accélération d'un grand nombre d'algorithmes de visualisation au moins pour le traitement des rayons primaires et d'ombrage. Bien que son adaptabilité aux caractéristiques puisse encore être améliorée les gains mesurés sur un ensemble de scènes représentatives ont prouvé la validité de la méthode. Ceux-ci varient, suivant les caractéristiques géométriques et optiques des objets, entre 60 et 99%.

- L'adaptabilité des algorithmes aux caractéristiques a permis d'améliorer la qualité de rendu liée généralement à un modèle à faces planes. Il est important que les algorithmes de visualisation ne soient pas monotypes. La possibilité de modifier en cours de traitement les algorithmes d'optimisation ou de rendu sera sans doute une nécessité pour les développements futurs. Le tracé de rayons n'étant qu'une méthode parmi d'autres pour la visualisation en CAO on peut imaginer que le concept d'adaptabilité peut être élargi à des notions de choix d'algorithmes ou de qualité de rendu.

Conclusion

Nous avons proposé avec la méthode de décomposition adaptative un principe d'analyse qui prolonge les travaux antérieurs sur les optimisations par décomposition. En implantant fortement la notion d'adaptabilité nous définissons une méthode d'analyse différente aussi bien dans la manière d'analyser le contenu d'une scène que dans le choix des techniques d'optimisation à mettre en oeuvre. Nous avons vu que la souplesse d'adaptation des algorithmes d'optimisation aux caractéristiques géométriques et optiques des objets permettait de simuler correctement des effets lumineux complexes tout en ayant une définition simple de la scène.

Plutôt qu'une décomposition statique de l'espace, utilisable pour l'étude des rayons primaires et secondaires, par exemple une décomposition de type octree, nous avons préféré une décomposition qui puisse s'adapter simplement à ce que pouvait voir l'observateur. En réalité, la décomposition adaptative sépare le traitement de ce qui est vu et de ce qui est éclairé. La décomposition associée aux différentes sources lumineuses est indépendante du point de vue, il n'est donc pas nécessaire de la recalculer, si, lors de la définition d'une animation, seul l'observateur est déplacé. Une décomposition spatiale ne permet pas une application sélective des différents types de rayons. En effet on est obligé pour chacun des rayons primaires, secondaires et d'ombrage, de rechercher une intersection en effectuant la traversée de la structure. Par contre, l'utilisation des projections d'objets permet de limiter le lancé de rayons aux strictes zones contenant de l'information. Il aurait été cependant très intéressant de pouvoir réaliser une comparaison chiffrée des deux approches.

La propagation d'une pyramide de visualisation a permis l'utilisation d'une décomposition surfacique pour optimiser le calcul des rayons réfléchis ou réfractés. Cependant la méthode connaît des limites liées essentiellement à la géométrie des objets et au coefficient de réfraction des objets transparents. Le mixage de méthode d'optimisation a permis d'apporter une solution satisfaisante à ce problème en effectuant automatiquement le choix de la méthode la plus adaptée aux contraintes de l'objet.

La réalisation a permis de valider les concepts et le bilan chiffré a confirmé l'intérêt de la méthode. Un comparatif des images a montré la fidélité des résultats obtenus par rapport à une méthode plus conventionnelle de traitement des rayons. Au-delà des performances chronométriques, le point le plus important reste le faible taux de rayons calculés inutilement. C'est-à-dire que nous avons été en mesure de cibler correctement les parties de l'image où se localisait l'information. L'application de l'algorithme de tracé de rayons ne s'est pas faite de manière aveugle, la conjonction d'une analyse "réfléchie" et adaptative de la scène ainsi que d'une application sélective d'algorithme favorise un accroissement important des performances.

Les applications futures devront sans aucun doute intégrer dans leur fonctionnalité la modularité et l'interactivité. Nous avons partiellement incorporé ces deux notions en essayant de développer un outil souple et convivial.

La modularité doit permettre une adaptation des algorithmes au type de scène traité. Il est difficilement concevable que les applications à venir ne soient pas en mesure de choisir l'algorithme le plus adapté à chaque étape de la réalisation d'une image de synthèse. Cette notion se retrouve dans la méthode de décomposition adaptative développée au LRIM. En effet, l'algorithme de visualisation choisit actuellement la méthode d'optimisation la mieux adaptée à l'objet en cours de traitement. Cette notion doit pouvoir facilement s'élargir à des concepts de plus haut niveau par exemple trouver le meilleur algorithme de visualisation en fonction du modèle de représentation des objets.

L'interactivité doit permettre à l'opérateur d'agir simplement sur les paramètres de calcul de l'image par exemple le modèle d'éclairage ou le niveau de réalisme recherché. L'interface que nous avons créée va dans ce sens en permettant à l'opérateur de choisir et de modifier de nombreux paramètres liés soit à la décomposition, soit à la forme d'optimisation ou encore au modèle d'éclairage.

Nos recherches futures vont s'articuler autour de deux points importants. Le premier consistera à augmenter la notion d'adaptabilité. La prochaine version de SACADO intégrera un modèleur hybride GSG /B-REP. Nous souhaiterions, après une analyse de la scène la plus automatique possible, que le module de visualisation choisisse les algorithmes les plus adaptés en fonction de caractéristiques fixées par l'opérateur comme les temps de réponse ou le degré de réalisme. Le second point, enfin, serait de porter la décomposition adaptative sur une machine à architecture parallèle. Associer à chacune des subdivisions un processeur devrait permettre d'améliorer les performances des algorithmes actuels. Cependant afin d'éviter les problèmes liés à la communication entre processus il semble plus souhaitable que chaque processeur ait une connaissance globale de la scène ou mieux une restriction du contenu à chaque sous-fenêtre. Les principes de décomposition et de propagation de faisceau semblent bien se prêter à une telle approche.

Références bibliographiques

Ouvrages de référence

AN INTRODUCTION TO RAY TRACING par Glassner A. éditions Academie press Londres

ALGORITHMES POUR L'INFOGRAPHIE par Rogers D.F. éditions Mac Graw Hill

COMPUTER GRAPHICS: Principles and practices par Foley J. Van Dam Feiner A. Hughes J.
éditions Addison Wesley

LA CFAO par Gardan Y. éditions Hermes Paris

LA SYNTHÈSE D'IMAGE par Peroche B. Argence J. Ghazanfarpour D. Michelucci D.
éditions Hermes Paris

Références bibliographiques

- [AMA 84] Amanatides J. "Ray tracing with cones" *Compt. Graph.* 18(3) 129-135 juillet 1984
- [AKI 91] Akimoto Takaaki, Mase kenji, Yasuhito Suenaga "Pixel-selected ray tracing" *IEEE Compt. Graph. & Appl.* 14-22 juillet 1991
- [APP 68] APPEL A. "Some techniques for shading machine renderings of solids" *AFIPS conf. proc.* Vol 32 37-45 1968
- [ARV 87] Arvo J., Kirk d. "Fast ray tracing by ray classification" *Compt. Graph.* 21(4) 55-64 juillet 1987
- [BOU 87] Bouatouch, Madini, Priol, Arnaldi "A new algorithm of space tracing using a CSG model" *Eurographics* 65-77 1987
- [BOU 88] Bouatouch, Priol, Arnaldi "subdivision spatiale et modélisation CSG pour le lancé de rayon" *Micad* 88 391-407 1988
- [BRO 84] Bronsvoort, Van Wijk, Jansen "Two methods for improving the efficient of ray tracing in solide modeling" *Compt. Aided Desing* 16 51-55 1984
- [CAT 74] Catmul E.E "A subdivision algorithm for computer display of curved surfaces" Ph.D. thesis, University of Utha Computer science departement Salt lake cit 1974
- [CRO 82] Crow F.C. "A more flexible image generation environment" *Computer Graphics* 16(3) pages 9 à 18 1982
- [CLE 88] Cleary j., Wyvill G. "Analyse of an algorithm for fast ray tracing using uniforme space subdivision" *The Visual Compt.* 4 65-83 1988
- [COH 85] Cohen M., Greenberg D.P. "The hemi-cube : A radiosity solution for complex environnement" *Computer Graphics* n° 19 vol 3 pages 31-40 Juillet 1985 SIGGRAPH'85
- [COH 88] Cohen M., Chen E., Wallace R. Greenberg D.P. "A progressive refinement approach to fast radiosity image generation" *Computer Graphics* n° 22 vol 4 pages 75-84 Aout 1988 SIGGRAPH'88
- [COH 92] Cohen D., Kaufman A., Yagel R. "Discrete Ray Tracing" pages 19 à 28 *IEEE Computer Graphics & Applications* septembre 1992
- [DEV 88] Deviller O., Puech C. Sillion F. "Efficiency of space subdivision structure for ray tracing" *Rapport de recherche* 88-2 laboratoire d'informatique de l'ENS 45 rue d'Ulm, 75230 Paris cedex 05 France mars 88
-

- [DEV 90] Deviller O., "The macro-région: an efficient space subdivision structure for ray tracing" Eurographics 27-39 1990
- [DIP 84] Dippe M., Swensen J. "An adaptative subdivision algorithm and parallele architecture for realistic image synthesis" Compt Graph vol 18 (3) 149-158 juillet 1984
- [EXC 87] Excoffier T., Tosan E. "une méthode d'optimisation pour le tracer de rayons" Micad 549-563 1987
- [FUJ 85] Fujimura K. Kunii T.L. "A hierarchical space indexing method" dans Kunii ed Compt. Graph.: Visual and art Poceeding of computer graphics tokyo'85 conference Springer Verlag 21-34 1985
- [FUJ 86] Fujimoto A. Tanaka T. Iwata K. "ARTS: accelerated ray-tracing systeme" IEEE Compt. Graph. Appl. 6(4) 16-26 Avril 1986
- [GAR 90] Gardan Y. Paul J.C. Vivian R. " Amélioration du calcul des images de synthèse par une analyse de la scène " GROPLAN 90 pages 1 à 13
- [GAR 91] Gardan Y. Lanuel Y. Vivian R. " Une décomposition adaptative pour le tracé de rayons : méthode et résultats " GROPLAN 91 pages 47 à 57
- [GEB 61] Gebhart B. "Heat transfert" Mc Graw-Hill New York 1961
- [GER 86] Gervautz M., "Three improvements of ray tracing algorithm for CSG trees" Compt. Graph. 10(4) 333-339 1986
- [GHA 92] Ghazanfarpour D. "visualisation réaliste par lancer de pyramides et subdivision adaptative" Micad 1992
- [GLA 84] Glassner A.S., "Space subdivision for fast ray tracing" IEEE Compt. Graph. & Appl. 4(10) 15-22 octobre 1984
- [GRE 86] Greenberg D.P., Haines E.A., "The light buffer: a shadows testing accelerator" Compt. Graph. & Appl. 6(9) 6-16 septembre 1986
- [GOR 84] Goral C., Torrance K., Greenberg D.P., Battaile B. " Modeling the interaction of light betwen diffuse surfaces" Computer Graphics n° 18 vol 3 pages 119-127 Juillet 1984 SIGGRAPH'84
- [HEC 84] Heckbert P.S., Hanrahan P., "Beam tracing polygonal objects" Compt. Graph. 18(3) 119-127 juillet 1984
- [HEG 87] Hegron G., "Dynamic management of 3D scenes" Eurographics '87 529-542
- [JAH 91] Jahami G. "Pour un système de synthèse d'image souple et évolutif" These de docteur de l'université de Saint Etienne 21 Mars 1991
- [JAN 86] Jansen F.W. "Data structures for ray tracing" dans data structure for raster graphics, proceeding workshop Eurographics seminars Springer verlag 1986 57-73
-

- [JEV 89] Jevans D., Wyvill B., "Adaptative voxel subdivision for ray tracing" *Compt. Interface '89* 164-172
- [JOY 86] Joy K.I. "Ray tracing parametric surface patches utilizing numerical techniques and ray coherence" *Siggraph '86* 279-285 1986
- [KAP 86] Kaplan M.R. "Space tracing a constany time ray tracer". *State of the art in image Synthesis*" (*Siggraph '85 course note*) vol 11 juillet 85
- [KAJ 82] Kajya J. T. "Ray tracing parametric patches" *ACM computer graphics* vol 16 N°3
- [KAY 79] Kay D., Greenberg D.P. "Transparency for computer synthesized images" *Computer Graphics* n° 13 vol 2 pages 158-164 Août 1979 SIGGRAPH'79
- [KAY 86] Kay T., Kajiya T. "Ray tracing complex scenes" *Siggraph '86* 269-278 1986
- [KUN 85] Kunii L.T. et Wyvill G. "CSG and ray tracing using fonctional primitives" dans *computer generated images Proceedings of graphics'* (Springer Verlag 137-152 1985
- [LAN 91] Lanuel Y. "Une méthode de lissage pour le tracé de rayons" *Laboratoire de recherche en informatique de Metz note interne*
- [MAU 91] Maurel H., Moisan B., Jessel J.P, Duthen Y., Caubet R., "Génération de séquences animées rendues par lancer de rayons" *GROS PLAN* 91 57-64 1981
- [NEM 86] Nemoto K., "An adaptative subdivision by sliding boundary surfaces for fast ray tracing" *Graphics interfaces* 43-48 1986
- [OHT 87] Ohta M., Maekawa M, "Ray coherence theorem and constant time ray tracing algorithm" *Compt. Graph. (Proc of CG international '87)* 303-314 1987
- [PUE 88] Puech C. Devillers O. "Une subdivision spatiale en deux étapes pour le tracé de rayon" *Rapport de recherche laboratoire d'informatique de l'ENS 45 rue d'Ulm, 75230 Paris cedex 05 France* 88
- [ROT 82] Roth S.D., "Ray casting for modeling solids" *Compt. Graph. & Image Proc.* 18 109-144 1982
- [SAH 89] Sahnoune M. "Contribution à l'intégration des surfaces gauches dans les modèles de solide" *Thèse de l'université de Metz décembre 89*
- [SAM 84] Samet H. "The quadtree and related hiérarchical data structure" *Computing Surveys* 16 juin 1984 187-260
- [SAM 89] Samet H. "Neighbor finding in images represented by octree" *CGVIP* 46(3) 367-386 juin 89
- [SED 84] Sederberg J.M., Anderson D.C., "Ray tracing for steiner patches" *Compt. Graph.* 18(3) 159-164 juillet 1984
-

- [SEW 86] Sweeney M.A.J and Bartels R.H "Ray tracing free-form b-spline surfaces" IEEE Cimpl. Graph. Appl 6(2) 41-49 Février 1986
- [SHI 87] Shinya M., Takahashi T. Naito S., "Principles and applications of pencil tracing" Compt Graph. 21(4) 45-54 juillet 87
- [SIL 89] Sillion F. "Simulation de l'éclairage pour la synthèse d'image : réalisme et interactivité" Thèse de doctorat en sciences Université de Paris sud centre d'orsay juin 1989
- [SNY 87] Snyder J.M. et Barr A.H. "Ray tracing complex models containing surface tessellations" Siggraph 87 119-128
- [THI 90] Thirion J.P "Data structures based on binary representation for ray tracing" Eurographics '90 531-541
- [TOR 90] Torres E. "optimisation of the binary space partition algorithm (BSP) for the sualisation of dynamic scene" Eurographics '90 507-541
- [VIV 91] Vivian R., Gardan Y. "Amélioration du calcul des images de synthèses par une analyse adaptative de la scène" Rapport de recherche 91(1) L.R.I. de Metz U.F.R M.I.M. ile du Saulcy 57000 Metz
- [VIV 91] Vivian R., Gardan Y. "Décomposition adaptative: méthode et résultats" Rapport de recherche 91(2) L.R.I. de Metz U.F.R M.I.M. ile du Saulcy 57000 Metz
- [VIV 93] Vivian R. Gardan Y. Lanuel Y. " Optimisation du tracé de rayons, une approche originale et adaptative située entre le ray-tracing et le beam-tracing " MICAD 93
- [WAL 87] Wallace J., Cohen M., Greenberg D.P. "A two-pass solution to the rendering equation: a synthesis of ray-tracing and radiosity methods" Computer Graphics n° 21 vol 2 pages 311-320 Juillet 1987 SIGGRAPH' 87
- [WAN 92] Wanger L., Ferwerda J., Greenberg D.P., "Perceiving Spatial Relationships in computer generated images" pages 44 à 55 IEEE Computer Graphics & Application Mai 1992
- [WAR 69] Warnock J.E., "A hidden surface algorithm for computer generated halftone pictures" technical report UTAH university june 1969
- [WEI 87] Weiller K. et Atherton P. "Hidden surface removal using polygon area sorting" Proceeding of Siggraph'87 214-222
- [WEG 84] Weghost H. , Hooper G. et Greenberg D.P. "Improved computational methods for ray-tracing" ACM TOG 3(1) 52-69 janvier 1984
- [WHI 80] Rubin S., Whitted T. "A 3-dimensional representation for fast rendering of complex scenes" Compt. Graph. 14(3) 110-116 juillet 80
- [WHI 80] Witted T. " An improved illumination model for shaded display" Communication of ACM n° 23 pages 343-349 1980
-

- [WYV 86] Wyvill G., Kunii T.L., Shira S.N, "Space subdivision for ray tracing in CSG"
IEEE Compt. Graph. & Applic. 6(4) 28-34 avril 1986

ANNEXE 1

LES MODELES UTILISES POUR LES TESTS

Nous présentons les modèles les plus importants utilisés pour valider la méthode. Les objets sont représentés en filaires avec une élimination partielle des faces cachées.

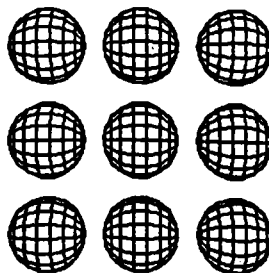
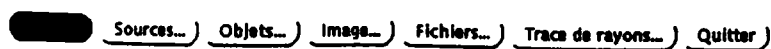
ENSEMBLE DES SPHERES

Nb Objets : 9

Nb Faces : 1296

Faces Concaves : NON

Faces Trouées : NON



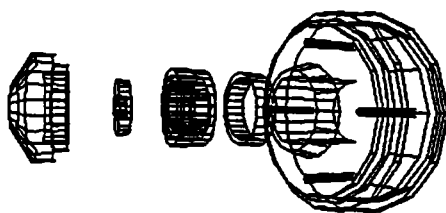
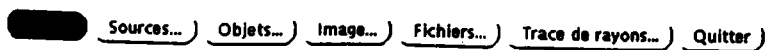
MOYEU DE RENAULT 5

Nb Objets : 6

Nb Faces : 612

Faces Concaves : NON

Faces Trouées : OUI



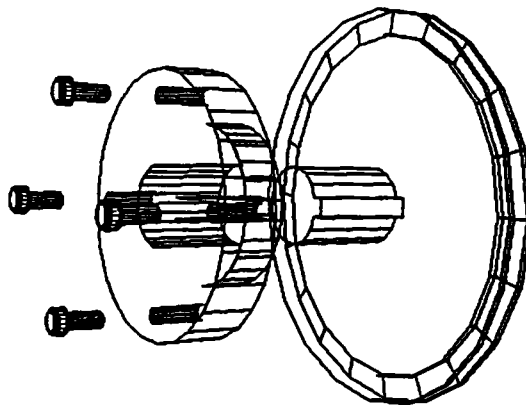
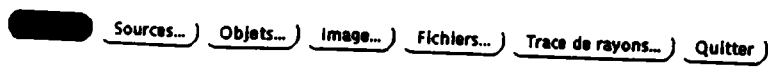
POULIE

Nb Objets : 9

Nb Faces : 1107

Faces Concaves : OUI

Faces Trouées : OUI



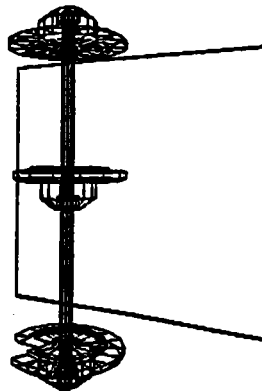
HALTERES IMPOSSIBLES

Nb Objets : 5

Nb Faces : 700

Faces Concaves : OUI

Faces Trouées : NON



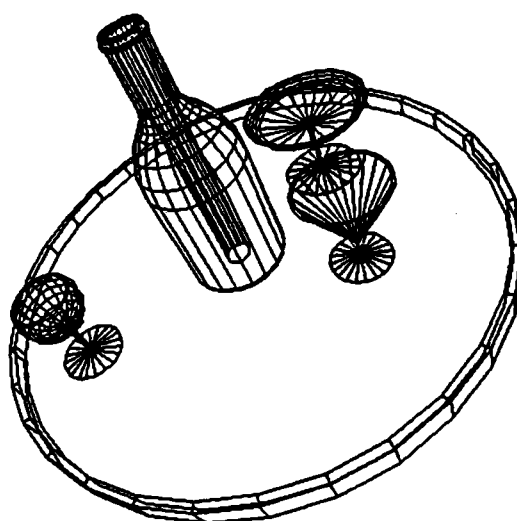
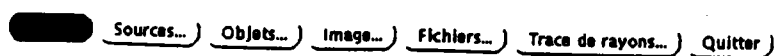
BOUTEILLE

Objets : 5

Faces : 2143

Faces Concaves : NON

Faces Trouées : NON



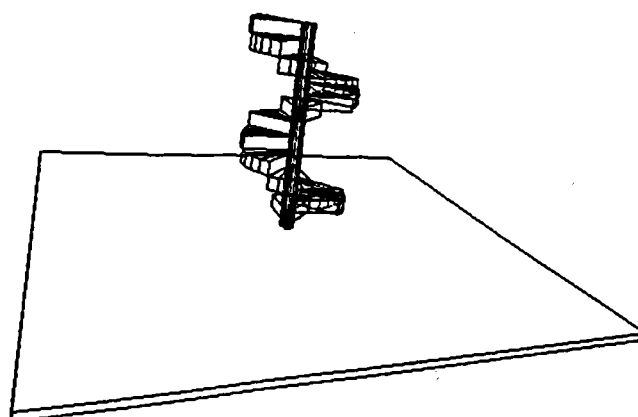
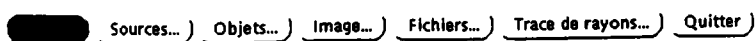
ESCALIER

Objets : 17

Faces : 294

Faces Concaves : NON

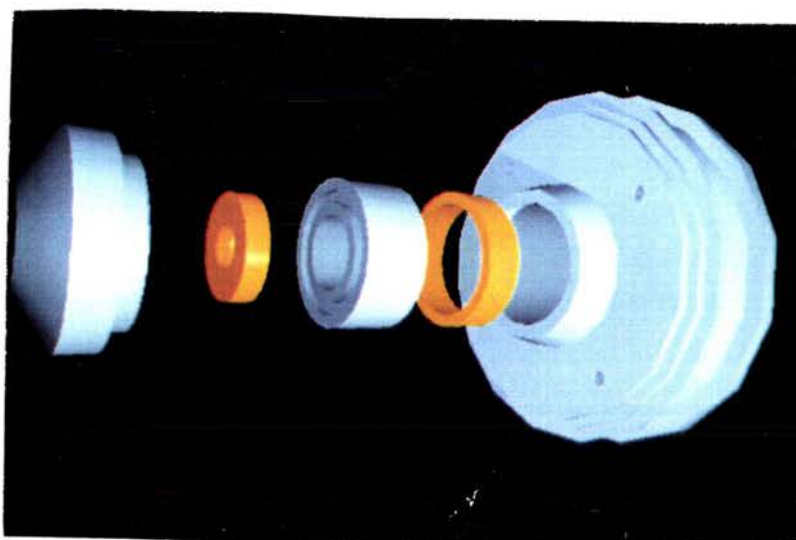
Faces Trouées : NON



ANNEXE 2

IMAGES DE REFERENCE

MOYEU DE R5 SANS EFFET OPTIQUE



Temps utilisateur

ENGLOBANTS : 8h 15'

DECOMPOSITION : 19' 30"

HALTERES REFLECHISSANTS



Temps utilisateur

ENGLOBANTS : 34h 45 '

DECOMPOSITION : 44'

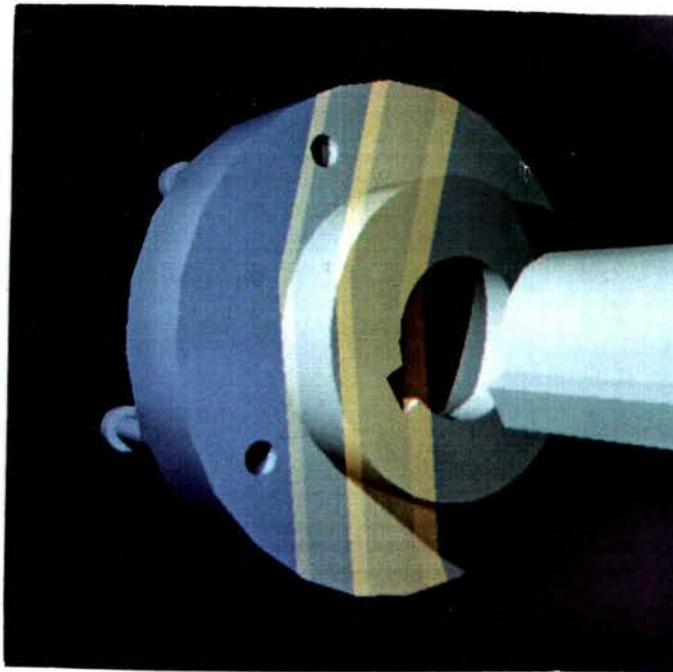
BOUTEILLE REFLEXIONS MULTIPLES



Temps utilisateur

ENGLOBANTS : 36h 55'

DECOMPOSITION : 5h 10'

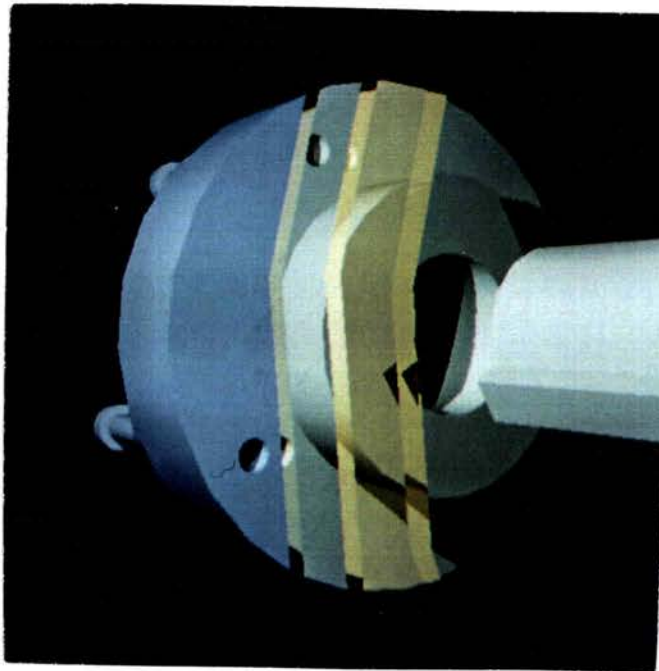
TRANSPARENCE AVEC COEF = 1

Temps utilisateur

ENGLOBANTS : 300h

DECOMPOSITION : 78h

TRANSPARENCE AVEC COEF $\diamond 1$



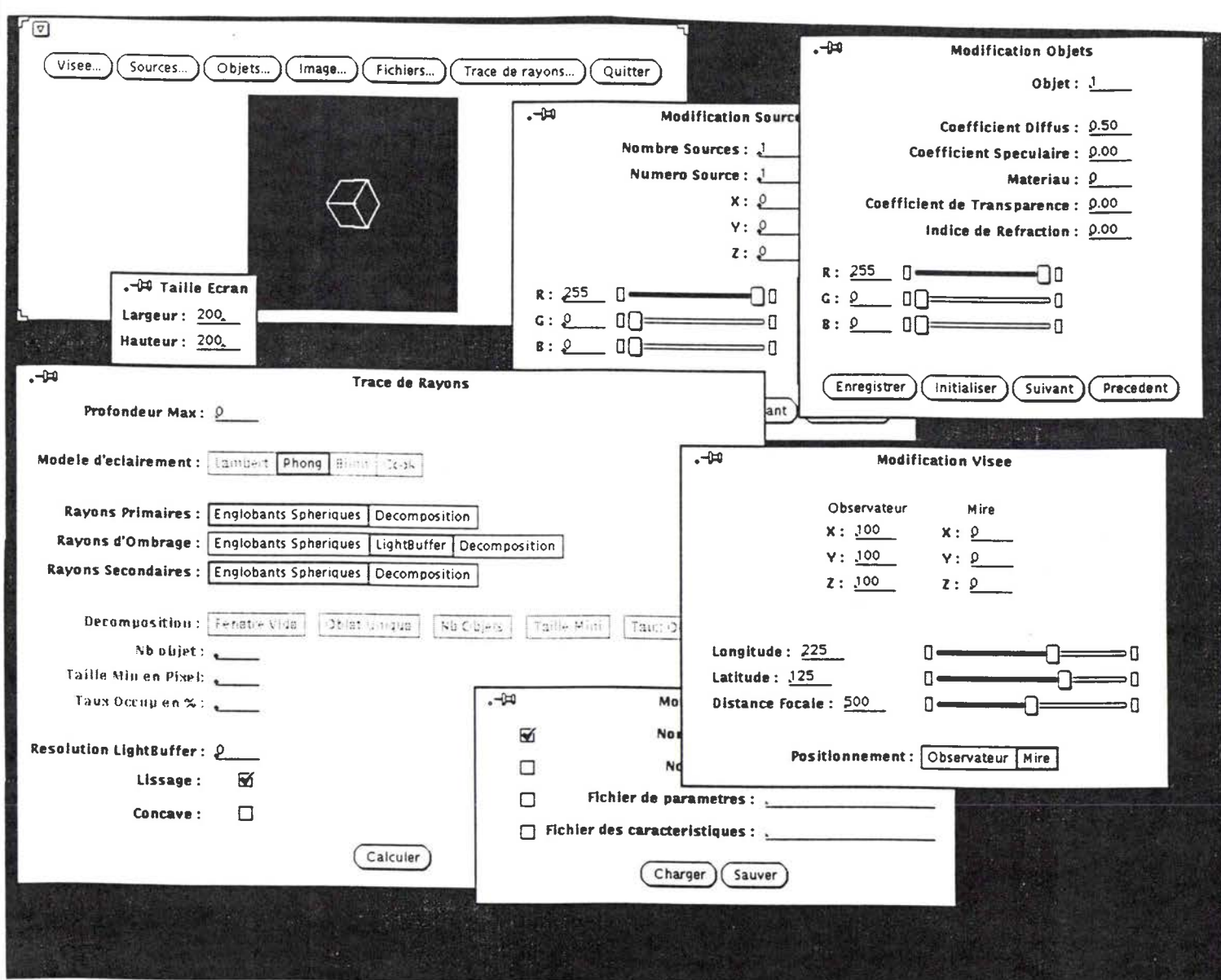
Temps utilisateur

ENGLOBANTS : 300h

DECOMPOSITION : 85h

ANNEXE 3

INTERFACE GRAPHIQUE DE L'APPLICATION



Il fait froid dans le scriptorium, j'ai mal au pouce. Je laisse cet écrit, je ne sais pour qui, je ne sais plus à propos de quoi: stat rosa pristina nomine, nomina nuda tenemus.

Umberto ECO
" le nom de la rose "

Résumé

Le tracé de rayons est une des techniques de synthèse d'images permettant de simuler des effets optiques complexes comme la réfraction et la réflexion. Malheureusement, cet algorithme reste très coûteux en temps de calcul. Afin de diminuer ces temps de traitements, nous proposons une optimisation basée sur une décomposition adaptative de l'espace de travail et une rationalisation des techniques d'optimisation. Une analyse du contenu de la scène et une localisation de l'information permettent une application sélective de l'algorithme de rendu. La méthode consiste en une analyse de la position et de la répartition des objets afin de déterminer des possibilités d'intersection. Une décomposition récursive et adaptative guidée par une étude du contenu permet de diminuer localement la complexité de la scène ce qui favorise une diminution des temps de traitement. Une propagation des résultats obtenus sur les rayons primaires et les rayons d'ombrage vers les rayons secondaires (réfléchis et réfractés) permet, sous certaines conditions, d'apporter une solution efficace au traitement des objets réfléchissants et transparents.