



HAL
open science

Modèle, calculs et applications de la visibilité en dimension n

Lilian Aveneau

► **To cite this version:**

Lilian Aveneau. Modèle, calculs et applications de la visibilité en dimension n . Synthèse d'image et réalité virtuelle [cs.GR]. Université de Poitiers, 2013. tel-00979543

HAL Id: tel-00979543

<https://theses.hal.science/tel-00979543>

Submitted on 16 Apr 2014

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

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

UNIVERSITÉ DE POITIERS
ÉCOLE DOCTORALE S2I
SCIENCES ET INGÉNIERIE POUR L'INFORMATION, MATHÉMATIQUES

HABILITATION à DIRIGER des RECHERCHES

Présentée et soutenue par

Lilian AVENEAU

Modèle, calculs et applications de la visibilité en dimension n

préparée au département SIC
de l'institut XLIM, CNRS UMR n°7252
soutenue le xx décembre 2013

Jury :

Rapporteurs

Kadi BOUATOUCH Professeur à l'Université de Rennes
Mathias PAULIN Professeur à l'Université Paul Sabatier
Christophe RENAUD Professeur à l'Université du Littoral

Examineurs

Djamchid GHAZANFARPOUR Professeur à l'Université de Limoges
Nicolas HOLZSCHUCH Directeur de Recherche INRIA
Pascal LIENHARDT Professeur à l'Université de Poitiers
Michel MÉRIAUX Professeur à l'Université de Poitiers

Table des matières

1	Introduction	1
1.1	De l'usage de la visibilité	1
1.2	Vers un nouveau modèle de la visibilité	3
1.3	Applications en synthèse d'images	4
1.4	Applications en simulation de propagation d'ondes	5
1.5	Plan	7
2	Introduction à la Visibilité Globale	9
2.1	Synthèse des travaux antérieurs	9
2.1.1	Partition des vues	10
2.1.2	Complexe et squelette de visibilité	10
2.1.3	Le calcul de Schubert en géométrie algorithmique	11
2.1.4	Calculs d'occultation fondés sur les coordonnées de Plücker	12
2.2	Calcul d'occultation en dimension 2	13
2.2.1	Espaces de droites en dimension 2	13
2.2.2	Calculer la visibilité en dimension 2	17
2.3	Vers un espace de droites en dimension 3	21
2.3.1	Les droites en dimension 3	21
2.3.2	Les coordonnées de Plücker en dimension 3	22
2.4	L'occultation dans l'espace de Plücker	28
2.4.1	Calcul d'occultation dans \mathbb{P}^5	28
2.4.2	Limites de ces méthodes	33
2.5	Conclusions	34
3	Contributions à la visibilité en dimension $n \geq 2$	37
3.1	Premiers travaux sur la visibilité	38
3.1.1	Calcul de visibilité en dimension 2	38
3.1.2	Calcul d'occultation en dimension 3	39
3.2	Introduction à l'algèbre de Grassmann	40
3.2.1	Présentation de l'algèbre de Grassmann	40

3.2.2	Propriétés et définitions utiles pour le calcul de visibilité	43
3.2.3	Algèbre de Grassmann sur un espace projectif \mathbb{P}^n	45
3.3	L'espace de droites en dimension n	48
3.3.1	L'espace \mathcal{D}_n et ses propriétés utiles	48
3.3.2	Le polytope minimal	53
3.4	Apports algorithmiques	54
3.4.1	Calcul d'occultation dans \mathcal{D}_3	55
3.4.2	Calcul à la demande	58
3.5	Conclusion	62
4	Travaux en Informatique Graphique	63
4.1	Travaux sur les ombres douces	63
4.1.1	Introduction au calcul d'ombre douce	63
4.1.2	Méthode à base de pré-calcul d'occultation	66
4.1.3	Méthode à base d'évaluation paresseuse	70
4.2	Autres applications de l'évaluation paresseuse	72
4.2.1	Réduction des biais dans les cartes de photons	72
4.2.2	Occlusion ambiante	78
4.3	Autres travaux en informatique graphique	83
4.3.1	Lobes de cosinus	83
4.3.2	Visualisation interactive de bâtiments	88
4.4	Conclusion	92
5	Travaux en Simulation de Propagation d'Ondes	95
5.1	Introduction à la simulation radio	95
5.1.1	Problématiques d'une simulation radio	95
5.1.2	La visibilité pour optimiser une simulation	99
5.2	Simulations en dimension 2	100
5.2.1	Graphe de visibilité en dimension 2	101
5.2.2	Exploitation du graphe de visibilité	103
5.3	Simulations en dimension 3	105
5.3.1	Nos deux simulateurs	105
5.3.2	Radars à travers les murs	110

5.3.3	La simulation en tunnels	115
5.4	La simulation pour l'optique sans fil	118
5.4.1	Nouvelle équation de transfert lumineux	119
5.4.2	Méthodes de Monte-Carlo	119
5.5	Conclusion	122
6	Conclusion & Perspectives	123
6.1	Conclusion	123
6.1.1	Modèles pour le calcul de visibilité	123
6.1.2	Synthèse d'images	124
6.1.3	Simulation de propagation d'ondes	125
6.2	Programme de recherche	127
6.2.1	Aspects formels	127
6.2.2	Perspectives en rendu	129
6.2.3	Perspectives en simulation de propagation d'ondes	132
A	Equation de transfert lumineux	135
A.1	Radiométrie	135
A.1.1	Angle solide	135
A.1.2	Grandeurs radiométriques	136
A.1.3	Propriétés	139
A.2	Fonction de distribution de réflectance bidirectionnelle	140
A.2.1	Présentation	140
A.2.2	Propriétés	141
A.3	Equation de transfert lumineux	142
A.3.1	Intégration sur l'hémisphère	142
A.3.2	Intégration sur les aires	143
A.3.3	Discussion sur l'équation de transfert lumineux	143
B	Méthodes de Monte-Carlo et tracé de chemins	145
B.1	Intégration de Monte-Carlo	145
B.1.1	Rappels de probabilité	145
B.1.2	Principe de base	149
B.1.3	Exemple : calculer π	149

B.1.4	La recherche de la vérité	151
B.1.5	Amélioration de la convergence	151
B.1.6	Roulette russe	152
B.2	Tracé de chemins	153
B.2.1	Méthode de base	153
B.2.2	Calcul avec estimation du prochain événement	154
C	Estimation de densité et carte de photons	157
C.1	Estimation de densité	157
C.1.1	Présentation générale	157
C.1.2	Estimateur naïf	157
C.1.3	Estimateur par noyaux	159
C.1.4	Biais dans l'estimation de densité	161
C.1.5	Estimation en dimension supérieure	161
C.2	Cartes de photons	162
C.2.1	Introduction	162
C.2.2	Détail de la méthode	163
C.2.3	Limitations	164
D	Pré-calculs de transfert énergétique	169
D.1	Bases de fonctions	169
D.1.1	Bases de fonctions orthonormales	169
D.1.2	Utilisation d'une base de fonctions orthonormales	170
D.1.3	Application avec les Harmoniques Sphériques	170
D.2	Méthodes PTE	174
D.2.1	Pré-calcul et stockage	176
D.2.2	Limitations	176
	Bibliographie	179

Introduction

Sommaire

1.1	De l’usage de la visibilité	1
1.2	Vers un nouveau modèle de la visibilité	3
1.3	Applications en synthèse d’images	4
1.4	Applications en simulation de propagation d’ondes	5
1.5	Plan	7

Ce mémoire d’habilitation à diriger des recherches condense les différents travaux que nous¹ avons menés ces 13 dernières années au sein du laboratoire SIC² autour de la visibilité. Ce chapitre d’introduction débute (*cf.* §1.1) naturellement par une discussion sur l’utilisation de la visibilité aussi bien en synthèse d’images, qui reste mon domaine de recherche principal, qu’en simulation de propagation d’ondes. Il se poursuit (*cf.* §1.2) par une justification de la nécessité d’un modèle unique pour traiter de la visibilité en dimension $n \geq 2$. Puis il présente succinctement nos différentes applications de la visibilité, autant dans le domaine de l’informatique graphique (*cf.* §1.3) qu’en simulation de propagation d’ondes (*cf.* §1.4). Enfin, il propose le plan de ce mémoire.

1.1 De l’usage de la visibilité

L’objectif de cette section est de montrer que la visibilité est un problème récurrent dans bien des disciplines, et non pas seulement en informatique graphique. Mais avant cela, il convient de définir correctement ce que nous entendons par “visibilité”. En pratique, il s’agit d’un concept polyvalent, apparaissant par exemple sous forme de :

- *requête*, il décrit la géométrie vue depuis un point, une surface ou un volume ;
- *prédicat*, il indique simplement si deux points A et B sont en visibilité mutuelle, *i.e.* si aucun objet géométrique n’est traversé par le segment de droite reliant A et B ;
- *prédicat* encore, mais pour deux ensembles de points, où il est question de l’existence d’au moins un couple de points de ces deux ensembles en visibilité mutuelle.

1. J’utilise le terme “nous” car il s’agit bien entendu d’un travail collectif, mené avec des stagiaires de DEA puis de master recherche, des doctorants, et des collègues de Poitiers et d’ailleurs.

2. Laboratoire ou département, car au fil des années le SIC fût un département du laboratoire IRCOM, un laboratoire FRE puis EA, et enfin un département de l’institut XLIM.

Nous retenons cependant un aspect commun, la *notion de droite*. En effet, autant pour décrire ce qui est vu depuis un point que pour les formes de type prédicat, chaque fois la visibilité revient à une question entre deux points, le long d'un segment de droite. C'est pourquoi tous les travaux antérieurs sur la visibilité reviennent systématiquement à travailler dans un espace de droites, plus ou moins bien défini.

La visibilité est un outil important dans bien des utilisations, à commencer par la synthèse d'images. Dans ce domaine, la visibilité est essentiellement utilisée pour le rendu, en commençant par les algorithmes classiques tels que l'algorithme du peintre, le tampon de profondeur, le lancé de rayons dans ses différentes formes (avec ou sans rebond), et les méthodes d'illumination globale (méthodes de radiosité, carte de photons, tracé de chemins, méthode de Metropolis, ...). À chaque fois, la question est de savoir ce qui est vu dans une direction donnée, le long d'un rayon primaire (partant de la caméra) ou secondaire (partant d'un point de réflexion). Dans les méthodes un peu évoluées simulant des sources lumineuses ou encore calculant l'illumination globale, des requêtes de visibilité sont aussi utilisées entre un point de la source et un point à éclairer, afin de simuler les ombres.

La visibilité est bien entendu utilisée aussi dans d'autres domaines que l'informatique graphique. Ainsi se retrouvent les mêmes besoins en simulation de propagation d'ondes radio, acoustiques ou autres. Lorsque de telles simulations sont basées sur la notion de rayon, alors la visibilité concernée est de type requête. La visibilité est aussi utilisée en amont de la simulation proprement dite, afin de réduire sa complexité : avec des réflexions uniquement spéculaires, pour un couple donné d'émetteur et récepteur, le nombre de chemins à tester est directement relié au nombre de surfaces ; par exemple, pour n surfaces et r réflexions, une méthode naïve va tester n^r chemins, *i.e.* vérifier l'appartenance de chaque point de réflexion à la surface, puis vérifier que ces points sont bien en visibilité mutuelle. En inversant le processus, une première étude de visibilité permet souvent de réduire de façon drastique le nombre de chemins à calculer lors de la simulation, que ce soit en tenant compte ou non de la position des émetteurs et récepteurs.

De la même façon et en informatique graphique, la visibilité trouve aussi des utilisations plus avancées que le simple rendu de base. Par exemple, les algorithmes de type tampon de profondeur nécessitent de projeter toute la géométrie sur la carte de pixels représentant l'image à rendre. Lorsque cette géométrie est complexe, et même avec les cartes graphiques actuelles, ce processus fini par coûter cher en terme de temps de calcul. Afin de limiter cette complexité, les méthodes à base de PVS³ sont utilisées : dans une phase initiale est effectué un calcul de visibilité pour tous les points appartenant à des volumes de l'espace bien choisis (les ensembles) ; lors du rendu, la géométrie utilisée pour le tampon de profondeur est réduite à celle potentiellement visible depuis le volume dans lequel est inclus la position de la caméra. Généralement, un calcul exact de la visibilité n'est pas obligatoire, une solution approchée mais *conservative* étant bien souvent suffisante. Ici, l'aspect conservatif signifie que les calculs de visibilité sont simplifiés, en prenant par défaut les primitives géométriques comme étant visibles, sauf lorsqu'il est possible de prouver facilement le contraire [Nirenstein 2003]. Néanmoins, une visibilité exacte est intéressante dans le sens où elle per-

3. *Potentially Visible Set*, pour ensemble (de primitives géométriques) potentiellement visibles.

met de réduire au maximum le nombre de primitives géométriques à fournir à l'algorithme de rendu.

1.2 Vers un nouveau modèle de la visibilité

Le calcul de visibilité en dimension 2 est maîtrisé depuis de nombreuses années. Les méthodes travaillent dans un espace projectif de droites, lui aussi de dimension 2. L'ensemble des droites traversant deux segments donnés est alors un polygone dans l'espace de droites, le *polygone initial*. Calculer l'occultation entre ces deux ensembles revient à supprimer à ce polygone initial l'ensemble des droites traversant les segments occultants. Ceci se traduit par des différences de polygones dans l'espace de droites.

En dimension 3, des travaux précédents ont essayé de généraliser ce qui fonctionne en dimension 2. Les seuls espaces de droites connus ou du moins utilisés sont basés sur les coordonnées de Plücker. Il s'agit d'espaces projectifs de dimension 5, mais dont les propriétés sont soit ignorées, soit non utilisées. Calculer la visibilité revient alors à effectuer des différences entre des polytopes dans l'espace de Plücker.

Malheureusement, de manière générale ces travaux calculent une visibilité approchée et non pas exacte. Pour commencer, les travaux antérieurs n'arrivent pas à définir la généralisation du polygone minimal, *i.e.* le *polytope minimal*. Ils donnent seulement un polytope suffisamment grand dans l'espace de droites pour contenir toutes les droites traversant deux polygones donnés. De plus, lorsque le plan support d'un polygone coupe l'autre polygone, alors la méthode achoppe ; il faut couper les polygones, puis les réduire ; ceci revient à biaiser les calculs, interdisant alors de parler d'une méthode exacte.

En résumé, cette méconnaissance de l'espace de droites en dimension 3 engendre des problèmes, interdisant une généralisation des méthodes fonctionnant en dimension 2 au cas de la dimension 3, sans parler des dimensions supérieures.

Nos travaux dans le cadre de la thèse de Sylvain Charneau (2007) nous ont permis de combler ce fossé, et de définir une méthode exacte de calcul de visibilité en dimension $n \geq 2$. En se reposant sur l'algèbre de Grassmann, nous avons formalisé l'espace de droites en dimension quelconque. Cette formalisation nous a permis de démontrer un théorème traitant de l'ensemble convexe des droites traversant deux faces en dimension n . Ce théorème conduit directement à une méthode de calcul du polytope minimal, qui reste exacte y compris dans les cas dégénérés. De plus, nous avons proposé plusieurs résultats théoriques, tels que l'orientation de deux droites en dimension quelconque et fonctionnant pour des droites réelles ou purement imaginaires, ou encore la distance entre deux droites conduisant à un calcul robuste.

À partir de ces résultats théoriques, nous avons défini deux modèles de calculs et de représentation de l'occultation. En pratique, ils reposent tous les deux sur un arbre de visibilité, contenant l'ensemble de l'information sur l'occultation entre deux faces d'un espace géométrique de dimension n . Le premier modèle permet d'obtenir un arbre de taille modeste, en choisissant les meilleures configurations possibles dans l'information de visibilité, no-

tamment en utilisant les silhouettes des *occulteurs*.

Notre second modèle est assez proche du premier, mais tente d'en corriger les défauts. Le premier d'entre eux concerne le temps de calcul d'un arbre de visibilité, qui peut devenir vraiment important. Le second défaut est que, dans une version allégée et plus rapide du calcul, le résultat peut devenir particulièrement gros en terme de mémoire occupée. Nous avons observé qu'une grande proportion d'un arbre de visibilité n'est jamais utilisée, soit en raison de nos applications, soit parce que l'information stockée ne concerne non pas des droites réelles de l'espace géométrique de dimension n , mais des droites imaginaires de l'espace de droites. Notre second modèle propose une évaluation paresseuse d'un arbre de visibilité. Lors d'une requête d'utilisation, les données manquantes sont calculées à la demande, évitant d'évaluer inutilement de l'information.

1.3 Applications en synthèse d'images

Notre premier domaine d'application de nos calculs de visibilité est la synthèse d'images, et plus précisément le rendu. La première application d'un calcul d'occultation exact concerne le rendu d'ombres douces, *i.e.* le calcul de l'éclairement induit par une source surfacique. Ces travaux furent conduit pendant la thèse de Frédéric Mora (2006). Nos arbres de visibilité, en versions pré-calculée ou paresseuse, sont utilisés de la façon suivante. Des deux faces concernées, l'une est une source lumineuse, l'autre une face quelconque à laquelle l'arbre est associé (lorsque plusieurs sources sont utilisées, plusieurs arbres sont attachés par face). Lors du rendu d'un point donné sur une face, nous utilisons le ou les arbres de visibilité associés. Depuis ce point, la visibilité est un sous-ensemble de l'information stockée dans l'arbre. Nous avons proposé un mécanisme simple d'extraction, particulièrement rapide y compris en version paresseuse. Le rendu repose ainsi sur une visibilité exacte de la source lumineuse depuis le point éclairé. Il est exact aussi pour des fonctions de réflectance de type Lambert et pour des sources uniformes. Cette méthode est plus rapide qu'un lancé de rayons, fût-il optimisé par paquets avec des instructions vectorielles.

Notre seconde application concerne le rendu de l'occlusion ambiante. Il s'agit d'une approximation de l'illumination indirecte, due à la géométrie voisine d'un point donné. Cette méthode assombrit un point d'autant plus qu'il a une géométrie dense et proche. Pour calculer exactement⁴ ce rendu, nous utilisons une version modifiée de nos arbres d'occultation. En effet, il nous faut une information de visibilité, autrement dit connaître la distance de la plus proche géométrie dans toutes les directions depuis un point donné. Avec nos arbres transformés, et une évaluation paresseuse, nous avons montré qu'il est possible d'obtenir un rendu de l'occlusion ambiante qui soit à la fois sans bruit et donc exact, mais aussi plus rapide qu'une méthode basée rayons. Bien sûr, nous sommes plus lents qu'une méthode GPU, mais cette dernière fournit des résultats particulièrement faux.

Une autre application de nos arbres de visibilité concerne la réduction des biais dans l'estimation de densité, par exemple pour le rendu à base d'une carte de photons. Elle a été dé-

4. L'occlusion ambiante est une approximation, dont nous proposons un calcul exact.

veloppée dans le cadre du doctorat de Romuald Perrot (2012). Nous avons analysé l'origine des biais dans les méthodes d'estimation de densité basée noyau. Leur origine est principalement un problème de définition du domaine d'application du noyau. Nous proposons d'utiliser des calculs de visibilité exacts, afin d'obtenir le domaine d'estimation exact sur lequel l'intégrale du noyau est ensuite calculée. Les résultats montrent une forte réduction du biais, plus ou moins rapide en fonction de la qualité ou de la méthode de rendu choisie.

Enfin, nous avons proposé deux autres méthodes, qui ne sont pas basées sur nos arbres de visibilité mais dans lesquelles la visibilité joue un rôle. Toujours dans le cadre de la thèse de Romuald Perrot, la première méthode concerne la réécriture de l'équation de transfert lumineux uniquement à l'aide de lobes de cosinus, *i.e.* de fonctions de la forme $S(\mathbf{o}) = s \max(\mathbf{a} \cdot \mathbf{o}, 0)^e$ dans laquelle s et e sont deux scalaires et \mathbf{a} et \mathbf{o} sont deux vecteurs unitaires. Ces lobes permettent d'exprimer la luminance émise depuis une surface, la luminance incidente depuis une autre, les fonctions de réflectance, mais aussi la visibilité. La principale difficulté réside en ce dernier terme ; nous avons proposé de reconstruire l'information de visibilité due à une géométrie donnée, en la projetant sur un masque. Notre méthode a été validée dans un démonstrateur interactif, permettant de visualiser des objets plus ou moins complexes avec des réflectances quelconques, et en utilisant plusieurs sources lumineuses. Cette technique permet de rendre dans des scènes dynamiques aussi bien des ombres douces ou dures, ce qui n'est pas le cas des méthodes précédentes à base de pré-calculs de transferts énergétiques.

Dans le cadre de la thèse de Maxime Maria démarrée fin 2012, notre dernière contribution concerne le rendu dans les grands bâtiments. Il s'agit d'une structure accélératrice pour la recherche des intersections les plus proches ou pour le calcul d'occultation. Notre approche ressemble à des *cellules et portails*, mais avec deux contributions principales. Pour commencer, notre structure accélératrice colle à la géométrie du bâtiment, car elle est automatiquement déduite depuis le modèle topologique stocké dans notre modéleur. Ensuite, elle utilise toutes les relations topologiques entre éléments de la scène, et non pas seulement celles entre les volumes. Plus précisément, nous utilisons les informations issues du modéleur topologique entre les sommets, les arêtes, et les faces. Ceci nous permet d'accélérer le parcours de la structure accélératrice, et d'obtenir des performances particulièrement intéressantes en résolution 1024×1024 , aussi bien sur CPU (340 images par secondes) que sur GPU (plusieurs milliers d'images par secondes).

1.4 Applications en simulation de propagation d'ondes

Nos travaux en simulation de propagation d'ondes trouvent leur origine dans la composition du laboratoire SIC, dans lequel je fus recruté en septembre 2000 après y avoir effectué mon doctorat soutenu en 1999. Ses membres appartiennent à trois sections du CNU : la 27, mais aussi les 61 et 63. Un des objectifs de mon recrutement était de renforcer les collaborations naissantes dans ce jeune laboratoire, construit en 1995. La simulation est un pont naturel entre les communautés synthèse d'images et propagation radioélectrique.

Dans la suite directe de mes travaux de thèse, dès mon recrutement j'ai continué à travailler sur un premier simulateur de propagation d'ondes radio, nommé CRT. Les simulations en question sont basées sur une méthode asymptotique, et donc sur une propagation le long de rayons directs, réfléchis, réfractés et diffractés. La caractéristique principale est que la réflexion est uniquement spéculaire, en raison des basses fréquences de la radio (par rapport à l'optique). De plus, le mécanisme de diffraction fait que les chemins de propagation sont *décidables*. La particularité des simulations que nous développons au laboratoire SIC est d'être exactes, engendrant l'ensemble des chemins. Le principal obstacle d'une telle méthode est sa complexité : pour n polygones d'une scène, et i interactions pour un chemin (*i.e.* réflexion et diffraction, la transmission étant à part), alors le calcul est en $\mathcal{O}(n^i)$.

Nos travaux concernent donc la réduction de la complexité de la simulation de propagation d'ondes radio. Nous développons dans ce mémoire ceux en relation avec la visibilité. La première méthode repose sur un lancé de faisceaux en dimension 3. Le mécanisme particulier de diffraction implique que nos faisceaux n'ont pas un bord plat mais conique, à la différence des travaux précédents [Ghazanfarpour 1998]. Si notre approche permet de réduire la complexité, elle ne suffit pas pour être complètement utilisable sur des configurations complexes. Le problème réside dans le traitement de la visibilité proprement dit.

Notre seconde approche traite la visibilité en utilisant un oracle. Ce dernier est un pré-traitement utilisant un lancé de rayons. Les rayons sont réfléchis, réfractés et diffractés dans la scène, conduisant à des potentiels chemins de propagation. Chaque face touchée par un chemin est marquée. Cette information de visibilité potentielle est utilisée dans la phase de simulation. Néanmoins une telle visibilité n'est pas exacte, et certaines contributions peuvent être ignorées dans la phase de lancé de rayons. Cette seconde méthode approchée offre cependant des performances très intéressantes, et est notamment utilisée dans plusieurs contrats industriels.

Dans le cadre de la thèse de Pierre Combeau (soutenue en 2004), nous avons étudié une méthode de simulation en dimension 2. Il s'agit d'un lancé de faisceaux, à la fois dans des plans horizontaux et verticaux. Utilisée comme une méthode de prédiction de visibilité et associée à des calculs en dimension 3, cette technique donne de très bons résultats sur de grands environnements.

Suite au succès de notre premier simulateur, nous avons décidé d'en écrire un nouveau en 2006 avec comme objectifs de simplifier la maintenance et plus spécialement les mises à jour, d'autoriser le travail collaboratif et d'être extensible. Ce simulateur nommé RaPSor a été utilisé notamment dans le cadre d'une thèse Cifre avec Alstom et le LEOST (à Lille), soutenue par Émilie Masson en 2010. Nous y avons travaillé sur des simulations dans des tunnels courbes, modélisés par des cylindres et des tores. Notre approche repose sur un lancé de rayons, associé à une méthode numérique pour affiner les chemins potentiellement reçus au niveau du récepteur.

Nous avons aussi utilisé RaPSor pour aider nos collègues du L3i de La Rochelle, qui travaille sur les radar ULB (*Ultra Large Band*). Nos simulations sont effectuées sur un très grand nombre de fréquences différentes, en dissociant la partie géométrique (pour une fréquence centrale) de son exploitation. L'application visée étant de la vision à travers les murs, nous

utilisons un modèle de parois de type multi-couches parallèles [Balanis 1989].

RaPSor est extensible, et permet aussi d'utiliser d'autres modèles physiques que la radio-électricité. Depuis 2010, nous travaillons sur la simulation de propagation de l'optique sans fil. L'objectif est de permettre les communications numériques en utilisant en émission des sources de type LED. Nous avons testé différents modèles de simulation, basés sur les méthodes de Monte-Carlo et de Metropolis. La comparaison des réponses impulsionnelles simulées avec celles issues de la littérature a permis de valider nos méthodes. Les simulations sont utilisées par nos collègues du département C2S2 de l'institut XLIM, à Limoges, comme base de travail de développement de nouveaux protocoles de communication.

1.5 Plan

Ce mémoire d'habilitation à diriger des recherches se poursuit suivant le plan suivant. Dans le chapitre 2 nous détaillons les travaux antérieurs sur la visibilité globale, en commençant par le problème en dimension 2, puis son extension en dimension 3. Notamment, nous décrivons les différentes lacunes des travaux précédents, principalement induites par un manque de connaissance de l'espace de droites. Dans le chapitre 3, nous détaillons nos modèles pour le calcul de visibilité, basés sur une définition formelle de l'espace de droites dans l'algèbre de Grassmann.

Les deux chapitres suivants sont dédiés aux applications, à commencer par l'informatique graphique (*cf.* chapitre 4) et la simulation de propagation d'ondes radioélectriques ou optiques (*cf.* chapitre 5).

Le chapitre 6 conclut nos travaux, et propose mon programme de recherche à court, moyen et long terme. Enfin, quatre annexes rappellent des éléments fondamentaux en informatique graphique, comme l'équation de transfert lumineux (*cf.* annexe A), les méthodes de Monte-Carlo (*cf.* annexe B), l'estimation de densité (*cf.* annexe C) et enfin les méthodes à base de pré-calcul de transferts énergétiques (*cf.* annexe D).

Introduction à la Visibilité Globale

Sommaire

2.1 Synthèse des travaux antérieurs	9
2.1.1 Partition des vues	10
2.1.2 Complexe et squelette de visibilité	10
2.1.3 Le calcul de Schubert en géométrie algorithmique	11
2.1.4 Calculs d'occultation fondés sur les coordonnées de Plücker	12
2.2 Calcul d'occultation en dimension 2	13
2.2.1 Espaces de droites en dimension 2	13
2.2.2 Calculer la visibilité en dimension 2	17
2.3 Vers un espace de droites en dimension 3	21
2.3.1 Les droites en dimension 3	21
2.3.2 Les coordonnées de Plücker en dimension 3	22
2.4 L'occultation dans l'espace de Plücker	28
2.4.1 Calcul d'occultation dans \mathbb{P}^5	28
2.4.2 Limites de ces méthodes	33
2.5 Conclusions	34

Ce chapitre propose un aperçu des travaux antérieurs sur la visibilité, en dimensions 2 ou 3. Il introduit les espaces de droites usuels en dimensions 2 et 3, notamment afin de montrer la relative absence de liens entre eux. Il insiste ensuite sur les calculs d'occultation dans un tel espace des droites, étant au coeur d'un grand nombre de nos travaux de recherche.

Ce chapitre termine par une justification d'une méthode ou d'un modèle unique pour traiter de la visibilité en dimension n .

2.1 Synthèse des travaux antérieurs

Nous présentons une synthèse des travaux existants sur la représentation globale de la visibilité, librement inspirée de Mora [Mora 2006] et de Durand [Durand 2000]. Cette synthèse présente également des travaux actuels sur l'application du calcul de Schubert à l'étude de la visibilité, qui partage quelques points communs avec notre approche présentée dans le chapitre 3.

2.1.1 Partition des vues

Les graphes d'aspects sont les premières techniques caractérisant la visibilité globale. Ils ont été introduits initialement en vision par ordinateur pour la reconnaissance d'un objet à partir des différentes vues possibles de celui-ci [Koenderink 1976, Koenderink 1979]. D'abord définie sur des objets polygonaux, leur construction fut par la suite étendue pour des objets lisses [Petitjean 1992].

Un graphe d'aspect est une représentation de toutes les vues d'un objet. Les vues sont regroupées en cellules dans un espace de vues représenté par la sphère des directions S^2 . Chaque cellule correspond à l'ensemble des vues pour lesquelles l'aspect de l'objet reste inchangé.

Les applications des graphes d'aspects sont assez limitées : d'abord, en raison d'un manque de robustesse pour la construction du graphe dans certains cas dégénérés ; ensuite parce que la taille du graphe en mémoire est très importante ; enfin, car il est particulièrement difficile d'en extraire une information pertinente.

Platinga et Dyer [Platinga 1990] proposent une structure, l'*asp* (pour *aspect representation*), dans le but de construire le graphe d'aspect d'un objet polyédrique. Ils utilisent une paramétrisation des droites ou des rayons de l'espace. Leur structure correspond, pour un objet donné, à une représentation de l'ensemble des droites traversant cet objet. Le calcul de l'occlusion d'un objet par un autre revient à soustraire l'*asp* de l'objet générant l'occlusion à celui de l'objet occulté. Malheureusement, l'*asp* est en pratique limité par une structure complexe et difficile à représenter, et à notre connaissance aucune implantation fonctionnelle n'existe.

Un problème récurrent apparaissant dans ces méthodes de représentation des vues des objets est l'incapacité d'en extraire certaines informations élémentaires concernant la visibilité. Par exemple, un graphe d'aspect ne permet pas de savoir si deux points sont mutuellement visibles. Ainsi, ces méthodes, bien que représentant une information de visibilité, ne suffisent pas à caractériser *toute* la visibilité.

2.1.2 Complexe et squelette de visibilité

Le complexe de visibilité en dimension 2 de Pocchiola et Vegter [Pocchiola 1993] apparaît comme la première structure permettant la description et l'étude de la visibilité globale dans le plan. Sa définition s'appuie sur une notion de rayons équivalents. Un rayon est considéré comme une demi-droite, ayant une origine et une direction. Par définition, les rayons équivalents ont une même droite support, et des origines mutuellement visibles.

Cette étude est étendue au cas tridimensionnel par Durand *et al.* [Durand 1997a]. La définition du complexe de visibilité en dimension 3 de Durand s'appuie sur la notion de segment libre maximal, *i.e.* un segment borné par deux objets et ne rencontrant aucun autre objet en son milieu. Cela correspond à un couple de points mutuellement visibles, chacun sur la surface d'un objet.

La construction du complexe de visibilité 3D de Durand n'est pas robuste aux dégénérescences et est particulièrement gourmande en ressources mémoire et de calcul. Goac [Goac 2004] a proposé un algorithme robuste de construction du complexe de visibilité 3D, notamment pour des objets courbes ou des polygones en positions quelconques. Son travail sur le complexe de visibilité repose principalement sur la paramétrisation de Plücker (*cf.* section 2.3.2). Goac note que ce formalisme est plus approprié que toute autre paramétrisation des droites de l'espace tridimensionnel, en raison de ses propriétés algébriques [Goac 2004]. Notamment, il lui permet de montrer des propriétés sur la complexité théorique du complexe de visibilité.

Pour pallier le problème de l'occupation mémoire du complexe de visibilité, Durand *et al.* proposent la structure de squelette de visibilité [Durand 1997b], une restriction du complexe de visibilité aux 0-faces et 1-faces (respectivement : les droites passant par deux sommets, et celles passant par un sommet et un point libre sur une arête). Si des algorithmes essaient d'améliorer sa construction [Duguet 2002, Zhang 2007, Lazard 2010], elle reste toujours bien délicate en pratique. Cette structure est plus effective, bien qu'elle représente une information incomplète de la visibilité. Cette simplification est toutefois suffisante pour son application en radiosité hiérarchique [Durand 1999a].

Le complexe de visibilité est avant tout une structure intéressante du point de vue théorique. Il a permis des avancées considérables dans l'étude du phénomène visuel en dimension 3, notamment sur la complexité probabiliste de ce phénomène : Goac montre ainsi que la taille du complexe croît linéairement en fonction du nombre d'objets [Goac 2004].

2.1.3 Le calcul de Schubert en géométrie algorithmique

Une branche de la géométrie algorithmique s'intéresse aux problèmes des droites tangentes à des objets, principalement des quadriques (*e.g.* [Sottile 2007a, Sottile 2002, Theobald 2002, Sottile 2007b]), et ce dans des espaces de dimension quelconque. Ce sont les seuls travaux à s'intéresser au phénomène visuel dans des espaces de dimensions supérieures à 3, avec des applications en mécanique ou en visualisation de données [Gritzmann 2005].

Leur approche fondamentale est celle du *calcul de Schubert* (*Schubert calculus*), une branche de la *géométrie énumérative*, introduite au 19^{ème} siècle par Hermann Schubert [Schubert 1879]. Ce calcul est utilisé dans un but théorique, afin d'étudier la complexité du phénomène visuel en dimension quelconque. Il consiste à déterminer le nombre fini de droites incidentes à un ensemble de variétés linéaires, à l'aide de structures combinatoires déduites d'équations sur des coordonnées de Plücker généralisées [Sottile 2003] (*cf.* section 2.3.2). Algébriquement, cela revient à énumérer le nombre de points d'intersection entre une variété, représentant les contraintes d'incidences à l'ensemble des variétés linéaires, et une grassmannienne, *i.e.* une variété correspondant à l'ensemble de droites dans un espace de droites (*cf.* section 2.3.2).

Ces travaux sont particulièrement intéressants, parce qu'ils reposent sur un formalisme *algébriquement* équivalent à celui que nous proposons dans le chapitre 3. Il s'agit en outre

des seuls travaux antérieurs utilisant une représentation générale des droites en toutes dimensions. Toutefois, leur point de vue est différent du nôtre, car ils ne s'intéressent pas à la représentation de la visibilité. Pour cette raison, leur approche ne nous satisfait pas directement.

2.1.4 Calculs d'occultation fondés sur les coordonnées de Plücker

Les travaux sur le calcul d'occultation dans l'espace des droites [Teller 1991a, Pellegrini 1993, Pu 1998, Bittner 2002, Nirenstein 2003, Mora 2006], utilisant la paramétrisation de Plücker, représentent la première solution algorithmique permettant de calculer puis d'utiliser dans un contexte applicatif une information de visibilité globale. Nos approches s'appuient essentiellement sur ces méthodes.

Les coordonnées de Plücker sont une paramétrisation analytique des droites de l'espace tridimensionnel dans un espace projectif \mathbb{P}^5 , appelé espace des droites ou encore *espace de Plücker* par abus de langage (cf. section 2.3.2). Elles procurent un cadre analytique pour faire des calculs sur des droites et des ensembles de droites. Il s'agit précisément du cas tridimensionnel des coordonnées de Plücker utilisées dans le calcul de Schubert.

Teller utilise la paramétrisation de Plücker afin de déterminer l'ensemble des droites passant par une suite de polygones [Teller 1991a]. Il applique ensuite cette méthode pour accélérer le calcul d'illumination dans des scènes intérieures [Teller 1994]. Ce travail constitue une première forme d'information de visibilité globale, mais ne permet pas immédiatement de calculer l'occultation, *i.e.* une représentation de la visibilité tenant compte de l'occultation d'un polygone par d'autres polygones.

Pellegrini [Pellegrini 1993] propose la construction d'un arrangement d'hyperplans dans l'espace de Plücker. Chaque hyperplan correspond à une arête (un bord) d'un polygone occultant la visibilité. Nous appelons de tels polygones des *occulteurs*. Le résultat est une partition de \mathbb{P}^5 en un ensemble de cellules, chacune représentant les droites passant par une même suite de polygones. Cet arrangement est étudié afin de déterminer une solution au problème du lancer de rayons en $\mathcal{O}(\log n)$, où n est le nombre d'hyperplans (d'arêtes de la scène). Cette solution reste cependant inexploitable en pratique, en raison du nombre de cellules dans l'arrangement : en effet, sa complexité est de l'ordre de $\mathcal{O}(n^4 \log n)$.

À partir de ces deux premiers travaux, Pu [Pu 1998] définit un premier calcul d'occultation à partir de la paramétrisation de Plücker. Il parvient alors à déterminer une représentation exacte de la visibilité de la scène depuis un polygone, en construisant un arrangement similaire à celui de Pellegrini. Toutefois, pour la même raison que précédemment, cette méthode s'avère inefficace en pratique, et limitée à des scènes contenant au plus une quinzaine de polygones.

Nirenstein *et al.* [Nirenstein 2002] et Bittner [Bittner 2002] réutilisent le même principe en l'améliorant, pour le calcul de PVS (ensemble des polygones visibles depuis une région de l'espace). La principale différence entre ces deux méthodes est l'information calculée et représentée : Bittner calcule une représentation complète de la visibilité de la scène depuis

une face polygonale, tandis que Nirenstein détermine seulement un prédicat d'existence de visibilité entre des couples de polygones. Si la méthode de Nirenstein est beaucoup plus rapide et légère en mémoire en pratique, elle ne constitue cependant pas une représentation de la visibilité globale.

Les travaux sur le calcul d'occultation reposant sur la paramétrisation de Plücker possèdent également des limites, concernant par exemple le traitement de certains cas dégénérés (cf. §2.4.2). De plus, d'un point de vue théorique ils n'ont pas permis d'améliorer la connaissance du phénomène de visibilité, ni de l'espace de droites permettant la représentation de ce phénomène. Enfin, ces méthodes, ainsi que les travaux sur le complexe de visibilité, ne semblent pas directement généralisables à toute dimension, bien que cela représente un intérêt, comme le montre l'étude du calcul de Schubert pour la visibilité.

2.2 Calcul d'occultation en dimension 2

En dimension 2, les méthodes pratiques de calcul d'occultation reposent sur la notion d'*espace dual* du plan affine¹, que l'on appelle par analogie l'*espace primal*. Cet espace défini, calculer la visibilité revient alors à manipuler des polygones convexes dans l'espace dual, chacun représentant un ensemble de droites visibles.

2.2.1 Espaces de droites en dimension 2

Dans la littérature coexistent beaucoup de définitions plus ou moins différentes d'un espace de droites de l'espace affine \mathcal{A}^2 . Ces définitions peuvent néanmoins être classées en deux catégories, en fonction de la dimension de l'espace de droites : un plan $2d$ ou un espace $3d$. Si les premiers semblent plus simples d'utilisation, il faut néanmoins souligner qu'ils souffrent de dégénérescences posant des soucis importants ; de plus, les premiers ne sont que des cas particuliers des seconds, généralement définis comme des espaces projectifs.

Dans un premier temps, nous présentons la définition analytique et affine, souvent rencontrée dans la littérature [Foco 2003]. Dans un second temps, nous présentons une définition algébrique de la dualité projective réelle dans un espace de dimension finie, telle qu'elle apparaît dans la littérature plus spécialisée.

2.2.1.1 Définition analytique d'un espace de droites

Soit l'espace affine \mathcal{A}^2 muni d'un repère cartésien (O, \vec{i}, \vec{j}) . Dans ce repère et pour toute valeur de x , une droite peut être donnée par une fonction de la forme $y = ax + b$, pour $(a, b) \in \mathbb{R}^2$. L'espace dual du plan affine est défini en associant à chaque droite $y = ax + b$ le couple (a, b) de \mathbb{R}^2 .

1. Le plan affine est l'association d'un espace de vecteurs et d'un espace de points géométriques.

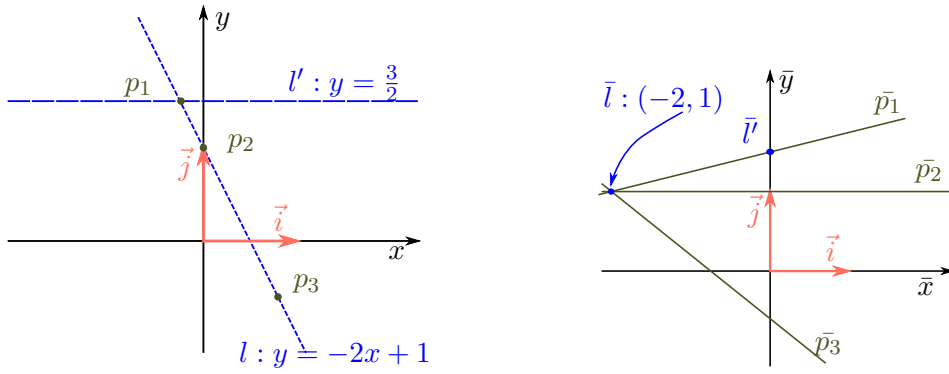
(a) Points et droites dans l'espace primal \mathcal{A}^2 .(b) Points et droites dans l'espace dual $\bar{\mathcal{A}}^2$.

FIGURE 2.1 – Espace dual $\bar{\mathcal{A}}^2$ du plan affine \mathcal{A}^2 . Cette dualité permet de caractériser les incidences des droites et des points dans l'espace primal.

Définition 2.1 L'espace dual $\bar{\mathcal{A}}^2$ du plan affine \mathcal{A}^2 , est un espace affine de dimension 2, défini par les couples (a, b) , dans un repère $(\bar{O}, \vec{i}, \vec{j})$, correspondant, dans \mathcal{A}^2 , à la droite d'équation $y = ax + b$.

D'après cette définition, le dual d'une droite de l'espace primal correspond à un point de l'espace dual. En outre, le dual d'un point (α, β) de l'espace primal est la droite d'équation $\bar{y} = -\alpha\bar{x} + \beta$ dans l'espace dual, relativement au repère $(\bar{O}, \vec{i}, \vec{j})$. Nous avons alors la propriété suivante.

Propriété 2.1 Si la droite d'équation $\bar{y} = -\alpha\bar{x} + \beta$, dans le repère $(\bar{O}, \vec{i}, \vec{j})$, est le dual du point (α, β) dans le repère (O, \vec{i}, \vec{j}) , alors chaque point (\bar{x}, \bar{y}) de cette droite correspond, dans l'espace primal, à une droite incidente au point (α, β) .

En effet, le point $(\bar{x}_0, -\alpha\bar{x}_0 + \beta)$ de la droite duale correspond par définition à la droite d'équation $y = \bar{x}_0x - \alpha\bar{x}_0 + \beta$ de l'espace primal. Posant $x = \alpha$, alors vient $y = \beta$. Le point (α, β) est donc un point de la droite dans l'espace primal.

La figure 2.1 est une illustration de droites et de points, d'une part dans l'espace primal (figure 2.1(a)) et d'autre part dans l'espace dual (figure 2.1(b)). À la droite l de l'espace primal correspond le point \bar{l} de l'espace dual. Dans l'espace primal, les points p_1 , p_2 et p_3 sont tous incidents à la droite l . Dans l'espace dual, ces mêmes points correspondent respectivement aux droites \bar{p}_1 , \bar{p}_2 et \bar{p}_3 , qui sont toutes incidentes au point \bar{l} . La dualité est bijective : ainsi, dans l'espace primal les deux droites l et l' sont incidentes au point p_1 ; dans l'espace dual, les deux points \bar{l} et \bar{l}' sont incidents à la droite \bar{p}_1 .

Le défaut principal d'une telle définition d'un espace de droites est qu'il ne permet pas de représenter les droites de la forme $x = x_0$ pour tout y , $x_0 \in \mathbb{R}$. Ajouter un point à l'infini ne résout pas complètement le problème, car il n'est pas possible de discriminer deux droites

$x = x_0$ et $x = x_1$. Afin de manipuler l'ensemble des droites du plan affine \mathcal{A}^2 , il faut donc utiliser un espace de droites projectif.

2.2.1.2 Dualité projective en dimension n

Nous rappelons ici le principe de dualité projective en dimension n quelconque [Delachet 1964].

De manière générale, la dualité projective consiste à associer bijectivement un point à un hyperplan dual et un hyperplan à un point dual. Dans le plan, cela correspond à associer les points aux droites, et inversement.

La notion de dualité projective est liée à la dualité linéaire ou vectorielle, définie dans l'espace vectoriel générant l'espace projectif. Pour cela, nous introduisons l'espace vectoriel $V \simeq \mathbb{R}^{n+1}$ et l'espace dual V^* des formes linéaires sur V .

À toute forme linéaire non nulle f de V^* est associé un hyperplan $\mathcal{H}_f = \{x \in V \mid f(x) = 0\}$ dans V . Inversement, à tout vecteur x de V est associé un hyperplan $\mathcal{H}_x^* = \{f \in V^* \mid f(x) = 0\}$.

Remarquons que les formes f étant linéaires, alors pour tout $\lambda \in \mathbb{R}^*$, nous avons $f(\lambda x) = (\lambda f)(x)$. Cela nous permet d'en déduire les deux relations suivantes :

$$\begin{cases} \mathcal{H}_f &= \mathcal{H}_{\lambda f} & \text{pour tout } f \in V^* \\ \mathcal{H}_x^* &= \mathcal{H}_{\lambda x}^* & \text{pour tout } x \in V \end{cases}$$

Ainsi, la dualité est valide à un réel non nul près. Cette dernière remarque nous conduit à la définition suivante de la dualité projective.

Définition 2.2 Soit \mathbb{P}^n l'espace projectif de dimension n , déduit de l'espace $V \simeq \mathbb{R}^{n+1}$. L'espace dual \mathbb{P}^{n*} de \mathbb{P}^n est l'espace projectif de dimension n déduit de V^* qui :

- à tout point X de \mathbb{P}^n , de représentant homogène x dans V , associe bijectivement l'hyperplan projectif déduit de l'hyperplan vectoriel \mathcal{H}_x^* de V^* , et
- à tout point F de \mathbb{P}^{n*} , de représentant homogène f dans V^* , associe bijectivement l'hyperplan projectif déduit de l'hyperplan vectoriel \mathcal{H}_f de V .

L'ensemble des points de \mathbb{P}^n incident à un hyperplan \mathcal{H} correspond, dualement, à l'ensemble des hyperplans incidents au point dual de l'hyperplan \mathcal{H} dans \mathbb{P}^{n*} , et inversement.

2.2.1.3 Définition projective d'un espace de droites

Dans ce paragraphe, nous appliquons la dualité projective en dimension 2 pour construire un espace de droites, permettant notamment d'illustrer le paragraphe précédent.

Autant dans \mathcal{A}^2 que dans \mathbb{P}^2 , une droite est en toute généralité un hyperplan de la forme $l : ax + by + c = 0$, correspondant à l'ensemble des points de coordonnées (x, y) . En posant $a' = -a/b$ et $b' = -c/b$, il apparaît que cet hyperplan correspond à la droite $y =$

$a'x + b'y$ selon la définition analytique précédente. Ainsi, à chaque point de $\bar{\mathcal{A}}^2$ correspond au moins un hyperplan de la forme $ax + by + c = 0$ dans \mathbb{P}^2 . Cependant, puisqu'elle englobe correctement les droites de la forme $x = x_0$ pour $y, x_0 \in \mathbb{R}$, cette nouvelle définition est exempte du défaut de la première.

Un tel ensemble admet une infinité de représentants : ainsi, pour $\lambda \in \mathbb{R}^*$ alors $\lambda ax + \lambda by + \lambda c = 0$ définit le même ensemble de points correspondant à la droite l . Autrement dit, une droite est définie à un scalaire *non nul* près. Puisque le triplet (a, b, c) est un représentant non unique d'une droite de \mathbb{P}^2 , alors il est évident qu'une droite de \mathbb{P}^2 correspond à une droite vectorielle de \mathbb{R}^3 , et est ainsi en bijection avec un point de \mathbb{P}^{2*} .

En appliquant le principe de la dualité projective, alors tout point de \mathbb{P}^2 est associé bijectivement à un hyperplan dans \mathbb{P}^{2*} .

2.2.1.4 Orientation des droites

Suivant [Stolfi 1991], ce même espace projectif nous permet de manipuler des droites orientées, en considérant non pas les droites vectorielles de \mathbb{R}^3 mais les demi-droites vectorielles. Par conséquent, le vecteur (a, b, c) de \mathbb{R}^3 est un représentant non unique de la droite orientée $\lambda(ax + by + c) = 0$, pour tout $\lambda \in \mathbb{R}^{+*}$. Changer l'orientation d'une droite correspond alors à changer le signe du triplet (a, b, c) . Dans l'espace \mathbb{P}^2 , cela revient à inverser le signe des deux demi-espaces séparés par l'hyperplan projectif correspondant à la droite.

Afin d'éclaircir cette notion d'orientation, prenons en exemple la droite orientée l qui passe par le point a de coordonnées (x_1, y_1) puis par le point b de coordonnées (x_2, y_2) , avec la contrainte que a et b soient deux points distincts. Cette droite est l'ensemble des points (X, Y) linéairement dépendants des deux précédents. En algèbre linéaire, c'est équivalent à dire que les trois vecteurs $(X, Y, 1)$, $(x_1, y_1, 1)$ et $(x_2, y_2, 1)$ forment une famille liée. Par conséquent, la matrice associée à la famille est de déterminant nul (elle est de rang 2) :

$$\begin{bmatrix} X & x_1 & x_2 \\ Y & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} = 0,$$

conduisant à l'équation suivante :

$$(y_1 - y_2)X + (x_2 - x_1)Y + (x_1y_2 - x_2y_1) = 0.$$

Il est clair que l'on vient de retrouver une droite sous forme d'un hyperplan projectif². Ce qui nous intéresse ici est l'orientation de la droite : considérant maintenant la droite l' passant par b puis par a . Le système est presque le même, sauf que les deux dernières colonnes sont inversées, et donc que la condition de famille liée donne l'équation suivante :

$$(y_2 - y_1)X + (x_1 - x_2)Y + (x_2y_1 - x_1y_2) = 0.$$

2. En pratique cette méthode produit les coordonnées de Plücker, et fonctionne en toute dimension $n \geq 2$ [Aveneau 2009].

Elle ne diffère de la première que par le signe : il s'agit bien de la même droite vectorielle dans le dual \mathbb{R}^{3*} , mais de la demi-droite vectorielle opposée selon l'orientation de Stolfi.

Soit le point c de coordonnées (x_3, y_3) . Calculer dans quel demi-espace se trouve c par rapport à l et à l' montre que ceux-ci sont de signes opposés :

$$\begin{aligned} l & : (y_1 - y_2)x_3 + (x_2 - x_1)y_3 + (x_1y_2 - x_2y_1) = \alpha, \\ l' & : (y_2 - y_1)x_3 + (x_1 - x_2)y_3 + (x_2y_1 - x_1y_2) = -\alpha. \end{aligned}$$

Ce résultat est tout aussi valide dans l'espace dual, pour lequel l'hyperplan dual du point géométrique définit deux demi-espaces, dans lesquels se trouvent des points duaux de droites géométriques.

Une autre façon de considérer les choses, primordiale pour le calcul de visibilité, est de dire que les points et droites orientées ont une orientation relative, fonctionnant aussi bien dans le primal que dans le dual.

2.2.2 Calculer la visibilité en dimension 2

La dualité projective définie, il est maintenant possible de traiter de la visibilité dans le plan \mathcal{A}^2 . Bien entendu, beaucoup de travaux antérieurs traitent de la visibilité en dimension 2, et il est donc impossible de tous les citer. Nous utilisons ici une référence récente, qui renvoie elle-même à des publications plus anciennes [Markovic 2010].

2.2.2.1 Droites traversant un segment de droite

Soit un segment de droite défini par ses deux extrémités a et b . Les droites incidentes au point a sont, dans l'espace dual, celles incidentes à la droite \bar{a} . Donc, les droites qui traversent le segment ab sont, dans l'espace dual, les points se trouvant entre les deux droites \bar{a} et \bar{b} . Cependant, et la figure figure 2.2 le montre clairement, entre les deux droites \bar{a} et \bar{b} se trouve l'ensemble des points de $\bar{\mathcal{A}}^2$. La dualité projective nous permet néanmoins de préciser les choses : chaque point de l'espace primal \mathcal{A}^2 devient un hyperplan dans l'espace dual $\bar{\mathcal{A}}^2$. Autrement dit, un point de l'espace primal sépare l'espace des droites en deux parties, l'une positive et l'autre négative. Chaque demi-espace correspond à un ensemble de droites *tournant* dans un certain sens autour du point associé dans l'espace primal.

Par exemple, sur la figure 2.2(a), les droites l et l' ont des orientations différentes, aussi bien pour le point a que pour le point b . La première, l , tourne dans le sens trigonométrique autour de a , et dans le sens horaire autour de b . Inversement, la droite l' tourne dans le sens horaire autour de a , et dans le sens trigonométrique autour de b . Chacune de ces deux orientations est associée à un demi-espace dans l'espace dual, respectivement négatif (trigonométrique) et positif (horaire). Par exemple, sur la figure 2.2(b), la droite l est dans le demi-espace négatif par rapport à l'hyperplan \bar{a} , et dans la partie positive par rapport à \bar{b} .

Attention : la figure 2.2(b) est en dimension 2, et suppose donc une certaine orientation des droites, à savoir de la gauche vers la droite (donc avec x croissant). Ceci est dû à la perte

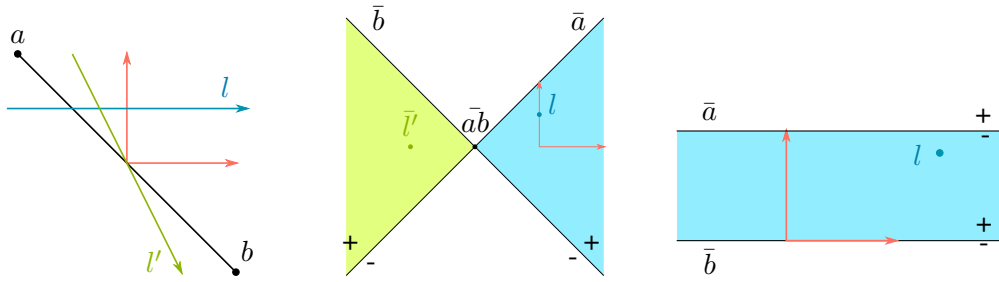
(a) Espace primal \mathcal{A}^2 .(b) Espace dual $\bar{\mathcal{A}}^2$.(c) $\bar{\mathcal{A}}^2$ après changement de repère.

FIGURE 2.2 – Ensemble de droites traversant un segment ab de l'espace primal \mathcal{A}^2 : (a) segment ab et droites le traversant ; (b) dans l'espace dual, on obtient un “papillon”, chacun de ses deux côtés correspond à un sens de traversée du segment ab ; (c) après changement de repère le point \bar{ab} est à l'infini.

d'une dimension lors du passage de \mathbb{P}^{2*} à $\bar{\mathcal{A}}^2$. Une façon simple pour se limiter à un seul côté du papillon est de changer de repère afin d'avoir la droite ab à l'infini : les deux droites duales des points a et b sont alors parallèles, et forment une bande contenant les droites traversant le segment ab (cf. figure 2.2(c)).

2.2.2.2 Droites traversant deux ensembles convexes

La clé du calcul de visibilité réside dans l'identification des droites traversant deux ensembles de points convexes de l'espace primal. En utilisant l'orientation des droites par rapport aux sommets des convexes dans le primal, alors l'ensemble des droites traversant ces deux convexes est défini dans le dual comme l'intersection de demi-espaces correctement orientés.

La figure 2.3 illustre une telle identification : dans l'espace primal (cf. figure 2.3(a)), les deux ensembles convexes sont les segments de droites ab et cd ; dans l'espace dual (cf. figure 2.3(b)), l'ensemble des droites traversant ab puis cd est le polygone de sommets \bar{ac} , \bar{ad} , \bar{bd} et \bar{bc} . Par exemple, la droite l traverse les deux segments : elle passe dans le sens trigonométrique autour de a et de c , et dans le sens horaire pour b et d ; dans le dual, l est dans les demi-espaces négatifs des hyperplans \bar{a} et \bar{c} , et dans les demi-espaces positifs des hyperplans \bar{b} et \bar{d} .

Le résultat est un polygone dont les sommets sont appelés *droites extrêmes* [Durand 1999b] : elles bordent l'ensemble des droites traversant les deux segments ab et cd . Il existe donc deux façons différentes mais complémentaires pour représenter les droites traversant deux ensembles convexes de points :

1. en utilisant les droites extrêmes,
2. par intersection de demi-espaces dans l'espace dual.

La seconde solution offre l'avantage de toujours fonctionner sans cas particulier (en terme

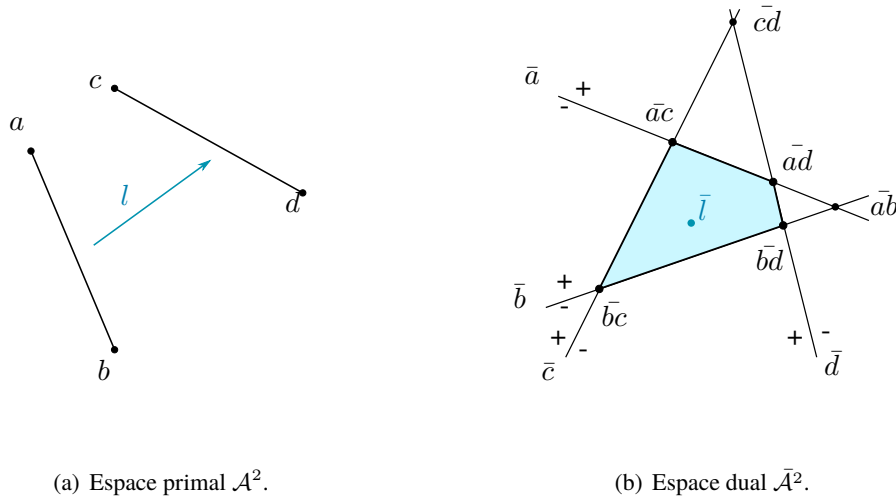
(a) Espace primal \mathcal{A}^2 .(b) Espace dual $\bar{\mathcal{A}}^2$.

FIGURE 2.3 – Visibilité entre deux segments de droite de l'espace primal (a) : dans l'espace dual (b), l'ensemble des droites traversant les segments ab puis cd est un polygone convexe dont les sommets sont les droites extrêmes ac , ad , bd , bc .

d'algorithme). La première semble intéressante, car elle donne une forme pratique du polygone représentant les droites traversant nos deux ensembles convexes. Cependant, si les deux géométries traversées ne sont pas simplement des arêtes mais des formes plus complexes, alors certaines droites extrêmes peuvent ne pas être pertinentes. La figure 2.4(a) propose un triangle abc et un segment de droites de . La figure 2.4(b) montre l'ensemble des droites traversant ces deux convexes : il s'agit des droites traversant les segments bc puis de ; ici les droites ad et ae n'ont aucune utilité, sinon de rendre le résultat plus difficile à interpréter : à proprement parler il ne s'agit plus de droites extrêmes.

2.2.2.3 Calculer la visibilité entre deux arêtes dans \mathbb{P}^{2*}

Nous noterons dans la suite $stab(\mathbf{A}_1 \dots \mathbf{A}_n)$, l'ensemble des droites passant par une suite de convexes $(\mathbf{A}_i)_{i=1\dots n}$ (i.e. des segments en dimension 2 ou des polygones en dimension 3). Notons que $stab(\mathbf{A}_1 \dots \mathbf{A}_n) = \bigcap_{i=1}^n stab(\mathbf{A}_i)$.

Soient un ensemble de convexes $(\mathbf{S}_i)_{i=1\dots s}$ et un ensemble d'occulleurs $(\mathbf{O}_j)_{j=1\dots o}$. Calculer l'occultation entre ces éléments consiste à représenter dans l'espace des droites l'ensemble $\bigcap_{i=1}^s stab(\mathbf{S}_i) \setminus \bigcup_{j=1}^o stab(\mathbf{O}_j)$, i.e. l'ensemble des droites traversant les convexes $(\mathbf{S}_i)_{i=1\dots s}$ mais ne traversant pas les occulleurs $(\mathbf{O}_j)_{j=1\dots o}$. Un tel calcul d'occultation se déroule en deux étapes :

1. Représenter l'ensemble $\bigcap_{i=1}^s stab(\mathbf{S}_i)$ des droites passant par les convexes \mathbf{S}_i .
2. Retrancher $\bigcup_{j=1}^o stab(\mathbf{O}_j)$, l'ensemble des droites traversant les occulleurs \mathbf{O}_j .

Prenons donc un exemple en dimension 2. À partir des représentations précédentes, le calcul de l'ensemble des droites traversant deux arêtes ab et cd de \mathcal{A}^2 sans traverser aucun obstacle entre temps suit le processus suivant :

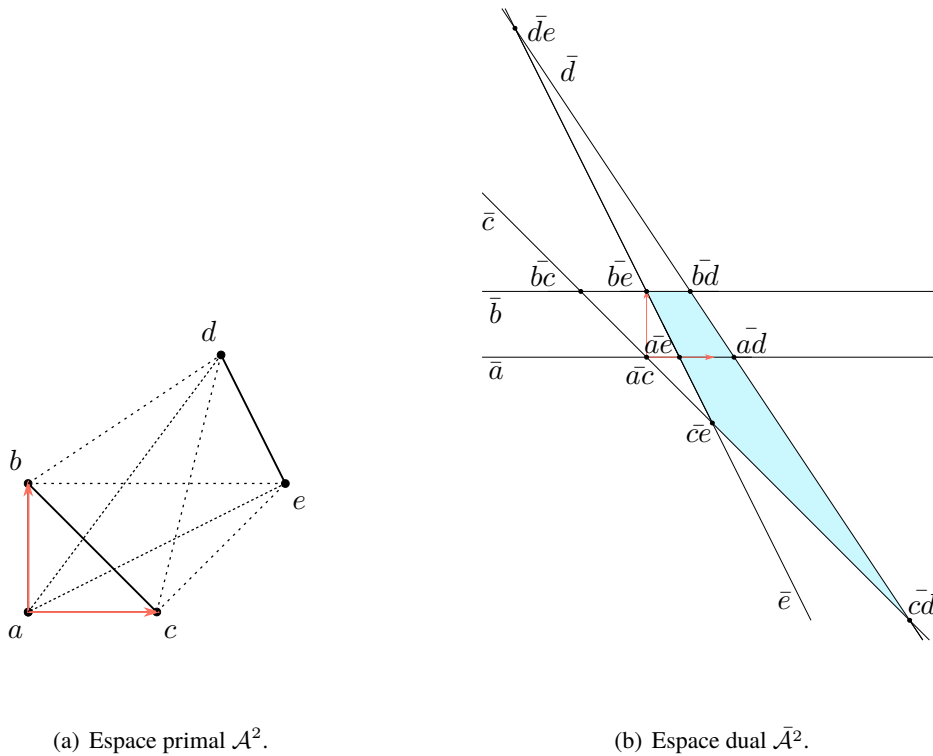
(a) Espace primal \mathcal{A}^2 .(b) Espace dual $\bar{\mathcal{A}}^2$.

FIGURE 2.4 – Visibilité entre un triangle et un segment de droite de l'espace primal (a) : dans l'espace dual (b), l'ensemble des droites traversant le triangle abc puis le segment cd est un polygone convexe dont les sommets sont les droites extrêmes be , bd , cd , ce .

1. Calculer l'ensemble de droites traversant ab puis cd : c'est le polygone $ac - ad - cd - cb$. Nous appelons cet ensemble *polygone minimal*, car c'est la plus petite représentation des droites traversant ab puis cd .
2. Pour chacun des occulteurs \mathbf{O}_i de bords p_i et q_i , bloquant potentiellement la visibilité entre ab et cd , soustraire du polygone minimal celui associé à l'ensemble des droites traversant ab puis p_iq_i . Cela revient à soustraire des morceaux du polygone minimal. Cette opération le transforme progressivement en un ensemble de polygones (\mathbf{P}_i) .
3. S'il reste un polygone dans (\mathbf{P}_i) , alors il existe encore entre ab et cd des droites non bloquées par les obstacles (\mathbf{O}_i) .

Ce principe est relativement simple à utiliser en dimension 2, même si les problèmes d'imprécisions numériques doivent être correctement traités, au risque de résultats erronés. Nous allons voir dans la suite de ce chapitre que son application en dimension 3 est loin d'être aussi simple.

2.3 Vers un espace de droites en dimension 3

Pour effectuer des calculs de visibilité en dimension 3, il est nécessaire de définir un espace de droites à l'image de l'espace de droites \mathbb{P}^{2*} , dans lequel une droite est connue par ses coordonnées de Plücker. Malheureusement, un tel espace en dimension 3 devient beaucoup plus délicat à manipuler que dans le cas précédent. Nous rappelons dans la suite la construction d'un espace de droites en dimension 3.

2.3.1 Les droites en dimension 3

En dimension 2, nous avons vu que l'espace des droites a la même dimension que l'espace géométrique (\mathbb{P}^2 et \mathbb{P}^{2*} étant de dimension 2). Malheureusement, en montant en dimension cette propriété n'est plus valide. Pour comprendre pourquoi, nous étudions la notion de degré de liberté d'une droite, en dimensions 2 puis 3. Nous introduisons ensuite les coordonnées de Plücker et l'orientation entre deux droites.

2.3.1.1 Degré de liberté en dimension 2

En dimension 2, une droite quelconque est représentée par un point, *i.e.* un sous-espace de dimension 0. Une interprétation de ce sous-espace considère que ce point est donc cette droite n'a aucun degré de liberté : une seule droite est représentée, celle passant par deux points distincts dans l'espace primal.

En réalité, il est possible de construire d'autres sous-espaces dans le dual, chacun lié à un degré de liberté différent. Considérons le cas d'une droite passant par un point fixe p , et par un second point libre p' sur une droite l donnée (telle que $p \notin l$) : une telle droite a donc 1 degré de liberté, puisque l est un sous-espace de dimension 1. Dans l'espace dual, cet ensemble de droites est un sous-espace de dimension 1, exprimant aussi ce degré de liberté : il s'agit donc d'une droite du dual. La section 2.2.1 nous a déjà appris qu'une telle droite du dual correspond à l'ensemble des droites du primal passant par un sommet donné ; ces deux conceptions sont en réalité équivalentes, car considérant le point à l'infini de la droite l on obtient bien toutes les droites passant par p , y compris la droite parallèle à l .

Il existe aussi des ensembles de droites ayant deux degrés de liberté. Là encore, plusieurs visions de ces ensembles cohabitent. Par exemple, en prenant deux droites parallèles et distinctes, alors deux points de l'espace (deux degrés de liberté) permettent de définir un tel ensemble ; en ne considérant qu'une droite l , alors un degré de liberté sur celle-ci plus une direction (un angle) permettent de définir le même ensemble. Dans l'espace dual, ces deux degrés de liberté correspondent à un sous-espace de dimension 2, et donc à un plan.

Le fait qu'il reste difficile de discriminer les droites parallèles à l est directement lié à la difficulté déjà évoquée de manipuler les droites de la forme $x = x_0$ pour tout $y, x_0 \in \mathbb{R}$, et donc la nature projective de l'espace dual de droites. Il manque pour représenter toutes ces droites une unique dimension. L'ensemble de toutes les droites de l'espace dual est donc de dimension 3.

2.3.1.2 Degré de liberté en dimension 3

Il semble naturel d'essayer d'étendre la notion d'espace de droites pour des espaces de dimensions supérieures à 2. Malheureusement cela est rendu très difficile en raison de la dimension d'un espace de droites, qui croît beaucoup plus vite que celle de l'espace géométrique. Pour bien s'en rendre compte, nous discutons ici du degré de liberté d'une droite en dimension 3.

Une droite passant par deux points distincts est toujours unique, et donc représentable par un sous-espace ayant 0 degré de liberté. Ainsi, un point semble être le représentant naturel d'une droite dans le dual, quelle que soit la dimension considérée.

L'ensemble des droites passant par un point p puis par une droite l est toujours représentable par un sous-espace de dimension 1, donc par une droite dans le dual. Bref, l'extension du cas en dimension 2 est là aussi direct.

Malheureusement, il existe beaucoup de nouveaux cas. Ainsi, l'ensemble des droites passant par deux autres droites ne contient plus toutes les droites de l'espace. Nous avons donc un sous-espace de dimension 2 dans l'espace de droites, sous-espace plus petit que l'espace de droites lui-même.

L'ensemble des droites passant par un premier point mobile sur une droite l , puis par un second point mobile lui aussi sur un plan Π est intrinsèquement de dimension $2 + 1 = 3$. Ainsi, un sous-espace de dimension 3 n'est-il pas l'espace de droites entier.

L'ensemble des droites passant deux points mobiles sur deux plans parallèles est de dimension 4. La direction d'un plan est de dimension 2 (et non pas 1 comme en dimension 2). Donc, pour représenter toutes les droites il faut un espace projectif associé à un espace vectoriel de dimension $4 + 2$, soit \mathbb{P}^5 .

2.3.2 Les coordonnées de Plücker en dimension 3

Un autre façon d'obtenir la dimension de l'espace de droites consiste à construire directement une droite en suivant le même raisonnement que dans la section 2.2.1.

2.3.2.1 Définition des coordonnées de Plücker

Dans un espace projectif de dimension 3, un point est représentable par 4 coordonnées homogènes. Une droite passant par deux points distincts est l'ensemble des points tels que le système suivant est lié [Aveneau 2009] :

$$\begin{pmatrix} X & x_1 & x_2 \\ Y & y_1 & y_2 \\ Z & z_1 & z_2 \\ 1 & 1 & 1 \end{pmatrix}$$

Autrement dit, tous les mineurs d'ordre 3 sont nuls :

$$\begin{bmatrix} X & x_1 & x_2 \\ Y & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} X & x_1 & x_2 \\ Z & z_1 & z_2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} Y & y_1 & y_2 \\ Z & z_1 & z_2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} X & x_1 & x_2 \\ Y & y_1 & y_2 \\ Z & z_1 & z_2 \end{bmatrix} = 0.$$

Cela correspond aux quatre équations suivantes :

$$\begin{aligned} X(y_2 - y_1) - Y(x_2 - x_1) - (x_1y_2 - x_2y_1) &= 0, \\ X(z_2 - z_1) - Z(x_2 - x_1) + (z_1x_2 - z_2x_1) &= 0, \\ Y(z_2 - z_1) - Z(y_2 - y_1) - (y_1z_2 - y_2z_1) &= 0, \\ X(y_1z_2 - y_2z_1) + Y(z_1x_2 - z_2x_1) + Z(x_1y_2 - x_2y_1) &= 0. \end{aligned}$$

Nous reconnaissons 6 termes différents dépendant du choix des deux points de la droite. Cela nous permet d'énoncer la définition suivante :

Définition 2.3 Soit la droite Π passant par le point de coordonnées (x_1, y_1, z_1) puis le point de coordonnées (x_2, y_2, z_2) . Les coordonnées de Plücker de Π sont définies de la façon suivante :

$$\begin{aligned} \pi_0 &= x_2 - x_1, & \pi_3 &= y_1z_2 - y_2z_1, \\ \pi_1 &= y_2 - y_1, & \pi_4 &= z_1x_2 - z_2x_1, \\ \pi_2 &= z_2 - z_1, & \pi_5 &= x_1y_2 - x_2y_1. \end{aligned}$$

Comme nous l'avons vu plus haut, l'espace de droites est un espace projectif de dimension 5 ; ainsi, les six coordonnées de Plücker correspondent-elles à un point d'un espace projectif, autrement dit à un vecteur d'un espace vectoriel. Néanmoins, les droites étant orientées, nous utiliserons ce vecteur avec les mêmes précautions que dans le système de Stolfi [Stolfi 1991], à savoir en ne considérant que les vecteurs colinéaires de même signe.

2.3.2.2 Orientation de deux droites en dimension 3

Les coordonnées de Plücker nous permettent aussi de discuter de l'orientation de deux droites, qui est un élément clé dans l'étude de la visibilité. Nous savons qu'une droite est définie à partir de deux points distincts. Soient la droite passant par les points p et q de coordonnées de Plücker $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5)$, et celle passant par les points a et b de coordonnées $(\pi'_0, \pi'_1, \pi'_2, \pi'_3, \pi'_4, \pi'_5)$.

Nous savons que le système construit par les 4 points p, q, a et b est de rang 2. Donc, le déterminant de la matrice formée par 4 sommets est nul :

$$\begin{bmatrix} x_p & x_q & x_a & x_b \\ y_p & y_q & y_a & y_b \\ z_p & z_q & z_a & z_b \\ 1 & 1 & 1 & 1 \end{bmatrix} = 0.$$

soit :

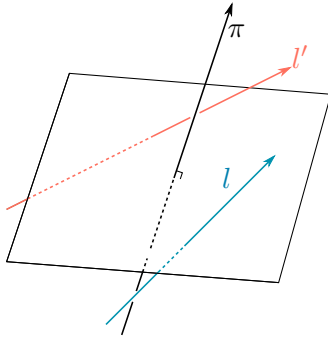
$$\pi_0\pi'_3 + \pi_1\pi'_4 + \pi_2\pi'_5 + \pi_3\pi'_0 + \pi_4\pi'_1 + \pi_5\pi'_2 = 0$$

La partie gauche de cette équation est appelée opérateur *side*.

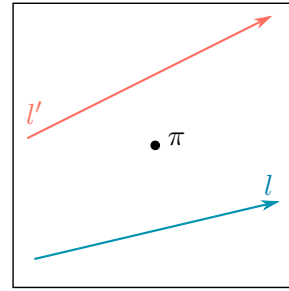
Définition 2.4 Soient deux droites Π et Π' . L'opérateur *side* est défini comme

$$\Pi \odot \Pi' = \pi_0\pi'_3 + \pi_1\pi'_4 + \pi_2\pi'_5 + \pi_3\pi'_0 + \pi_4\pi'_1 + \pi_5\pi'_2$$

Cet opérateur décrit l'incidence de deux droites : si elles sont libres, alors cet opérateur donne toujours une valeur non nulle. Dans le cas contraire c'est que les deux droites sont liées : soit elles se croisent et donc le système construit par les 4 points est de rang 3, soit elles sont confondues et donc le système est de rang 2. Dans les deux cas, l'opérateur est nul. Notons que si les deux droites sont parallèles, alors le système est de rang 3, et donc cet opérateur vaut 0 (équivalent à l'incidence à l'infini).



(a) Exemple de droites dans \mathcal{A}^3 .



(b) Droites projetées sur un plan orthogonal à π .

FIGURE 2.5 – Orientation de droites dans \mathcal{A}^3 : (a) les droites l et l' ne sont pas incidentes à la droite π ; (b) dans un plan orthogonal à π , les (projections des) droites l et l' tournent respectivement dans le sens trigonométrique et horaire autour de π (et inversement π autour de l et l').

Dans le cas d'un système libre, l'opérateur *side* prend une valeur non nulle. En pratique, il donne l'orientation relative des deux droites entre elles, avec une valeur positive si l'une tourne autour de l'autre de façon trigonométrique³, négative pour le cas horaire (cf. figure 2.5). Cette orientation est celle permettant de calculer la visibilité en dimension 3.

Propriété 2.2 L'opérateur *side* donne l'orientation relative de deux droites données. Soient les droites Π et Π' .

- $\Pi \odot \Pi' = 0$ si et seulement si les droites Π et Π' sont incidentes.
- $\Pi \odot \Pi' < 0$ si et seulement si les droites Π et Π' tournent l'une par rapport à l'autre dans le sens horaire.

3. En dimension 3, l'orientation est inversée par rapport à celle en dimension 2.

- $\Pi \odot \Pi' > 0$ si et seulement si les droites Π et Π' tournent l'une par rapport à l'autre dans le sens trigonométrique.

Notons que l'opérateur *side* revient à considérer une droite comme un hyperplan dans \mathbb{P}^5 . En effet, pour un point donné (une seconde droite), alors l'opérateur *side* indique de quel côté se trouve ce point par rapport à l'hyperplan associé à l'autre droite.

Définition 2.5 Soit la droite l de \mathcal{A}^3 , dont les coordonnées de Plücker Π_l sont connues. Dans l'espace de Plücker, l est associée à un hyperplan noté Π_l^\perp , dont les coordonnées sont :

$$(\pi_3, \pi_4, \pi_5, \pi_0, \pi_1, \pi_2).$$

2.3.2.3 Reconnaître un droite dans \mathbb{P}^5 , notion de Grassmannienne

Associer les coordonnées de Plücker à des droites vectorielles d'un espace vectoriel de dimension 6, support d'un espace projectif de dimension 5, ne permet pas pour autant de construire un *bon* espace de droites. En pratique, toutes les droites vectorielles (tous les points de \mathbb{P}^5) ne sont pas associées à des droites géométriques, mais seulement une petite partie qui forme un sous-espace de dimension 4. Ce sous-espace ne peut malheureusement pas être associé directement à un espace vectoriel, et n'est donc pas un espace projectif.

En pratique, un point de \mathbb{P}^5 est une droite géométrique si et seulement elle est sur la grassmannienne Ω ou encore *quadrique de Plücker*, une variété de \mathbb{P}^5 correspondant aux points π , telle que :

$$\pi_0\pi_3 + \pi_1\pi_4 + \pi_2\pi_5 + \pi_3\pi_0 + \pi_4\pi_1 + \pi_5\pi_2 = 0$$

Cette équation correspond à l'application de l'opérateur *side* sur une et une seule droite vectorielle. Autrement dit, un point Π dans \mathbb{P}^5 identifie correctement une droite géométrique si et seulement si la droite en question est incidente à elle-même, ou encore si et seulement Π appartient à Π^\perp .

Le fait que tous les points de \mathbb{P}^5 ne correspondent pas à des droites rend les calculs de visibilité en dimension 3 bien plus complexes qu'en dimension 2. En effet, pour un ensemble de droites défini par son bord (soit par ses sommets, soit par ses hyperplans), le fait qu'il ne soit pas vide ne suffit pas à conclure qu'il contient des droites. Pour cela, il faut aussi tester son intersection avec la grassmannienne.

2.3.2.4 Droites passant par un ou plusieurs polygones convexes

Teller *et al.* définissent une méthode algorithmique pour caractériser l'ensemble des droites passant par un ou plusieurs polygones [Teller 1991a].

Ce calcul repose sur l'opérateur *side*. Soit un triangle \mathbf{T} (ou tout autre polygone convexe), d'arêtes e_1 , e_2 et e_3 (cf. figure 2.6(a)). Les arêtes de T sont orientées de sorte que la seconde extrémité d'une arête coïncide avec la première extrémité de l'arête suivante. Alors toutes

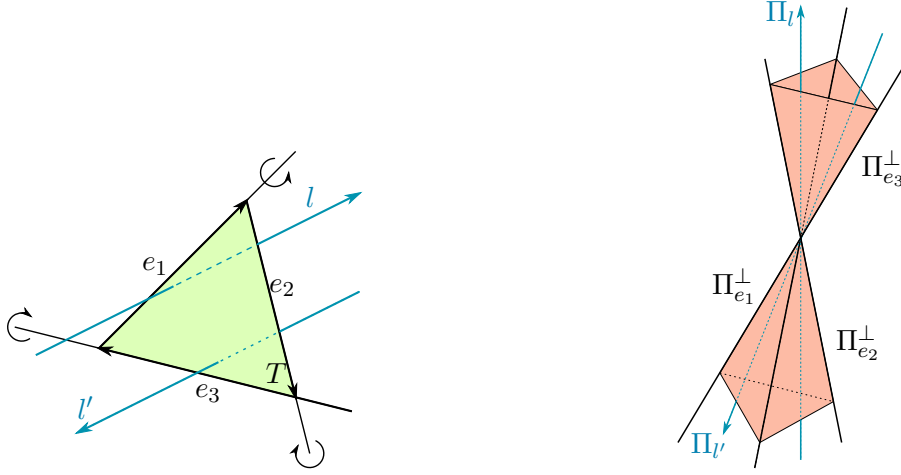
(a) Droites dans \mathcal{A}^3 traversant un triangle T .(b) Cône vectoriel dans l'espace de Plücker (attention, toute illustration dans \mathbb{P}^5 est trompeuse).

FIGURE 2.6 – Droites passant par un triangle T : (a) illustration dans l'espace \mathcal{A}^3 ; (b) dans l'espace de droites, l'orientation consistante des arêtes de T permet de définir un cône vectoriel, dont l'intérieur contient toutes les droites traversant T .

les droites traversant le triangle sont orientées de la même manière relativement à chaque arête du triangle.

Soit Π_{e_i} le point de \mathbb{P}^5 représentant la droite support de l'arête e_i dans \mathbb{P}^5 . Une droite l traverse le triangle T si et seulement si elle vérifie la propriété suivante :

$$\text{signe}(\Pi_{e_1} \odot \Pi_l) = \text{signe}(\Pi_{e_2} \odot \Pi_l) = \text{signe}(\Pi_{e_3} \odot \Pi_l) \quad (2.1)$$

Selon le sens de traversée du triangle par la droite l , le signe de $\Pi_{e_i} \odot \Pi_l$ sera positif ou négatif pour tout i dans $\{1, 2, 3\}$ (cf. figure 2.6(b)).

En considérant les hyperplans $\Pi_{e_i}^\perp$, l'équation (2.1) signifie que toutes les droites passant par le triangle appartiennent à l'intersection des demi-espaces positifs $\bigcap_{i=1}^3 (\Pi_{e_i}^{\perp+})$, ou négatif $\bigcap_{i=1}^3 (\Pi_{e_i}^{\perp-})$, selon le sens de traversée du triangle. Ces intersections définissent deux «cônes» opposés dans l'espace \mathbb{R}^6 déduit de \mathbb{P}^5 (cf. figure 2.6(b)). Ces deux cônes correspondent projectivement à un *polytope convexe* dans \mathbb{P}^5 .

Définition 2.6 Un *polytope convexe*⁴ est un ensemble défini dans un espace de dimension n , soit par l'enveloppe convexe d'un ensemble fini de points, soit par l'intersection d'un ensemble fini de demi-espaces. [Boissonnat 1998]

4. Dans la suite, le terme «polytope» fait toujours référence aux ensembles convexes représentant la visibilité dans l'espace des droites, tandis que le terme «polygone», *i.e.* un polytope de dimension 2, fait référence aux faces de l'espace géométrique entre lesquelles la visibilité est calculée.

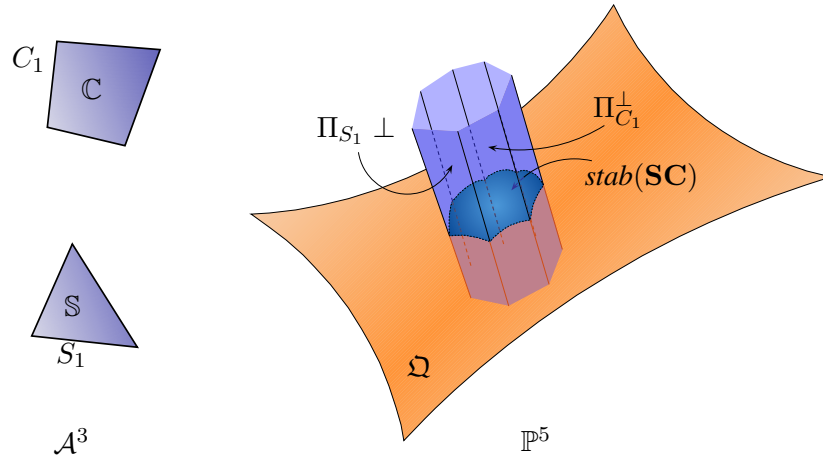


FIGURE 2.7 – Ensemble convexe déterminé, dans \mathbb{P}^5 , par les hyperplans associés aux arêtes de deux polygones S et C de l'espace géométrique. Cet ensemble est sous-contraint, signifiant que la donnée des hyperplans associés aux arêtes ne suffit pas à déterminer une représentation par sommets du polytope.

Comme en dimension 2, cette caractérisation des droites traversant un polygone se généralise à plusieurs polygones convexes $(S_i)_{i=1\dots s}$, en utilisant toutes les arêtes $(e_j)_{j=1,\dots,p}$ orientées des polygones. Le résultat est un polytope défini par l'ensemble des points Π dans \mathbb{P}^5 , vérifiant l'équation :

$$\forall j \in [1; p], \Pi \odot \Pi_{e_j} \geq 0$$

Teller remarque que l'ensemble déduit de l'intersection des demi-espaces positifs associés aux hyperplans Π_{e_j} n'est pas toujours borné, tandis que l'intersection du polytope avec la grassmannienne Ω l'est toujours. L'intersection de Ω avec le polytope correspond alors aux points représentant les droites $\bigcap_{i=1}^s \text{stab}(S_i)$ passant par l'ensemble des polygones $(S_i)_{i=1\dots s}$. Le polytope est toujours borné car l'espace projectif \mathbb{P}^5 est géométriquement fermé. Toutefois, il n'existe pas toujours de V -représentation du polytope (cf. figure 2.7).

Définition 2.7 *La description d'un polytope convexe par l'ensemble de ses sommets, i.e. le plus petit ensemble des points dont le polytope est l'enveloppe convexe, est appelée **V-représentation** du polytope.*

*De manière équivalente, la description d'un polytope convexe par les hyperplans de son bord, i.e. le plus petit ensemble d'hyperplans délimitant les demi-espaces dont le polytope est l'intersection, est appelée **H-représentation** du polytope.*

D'un point de vue algorithmique et pour des raisons d'efficacité, il est préférable de posséder une V -représentation du polytope. En notant que seule l'intersection du polytope et de la quadrique Ω est intéressante pour le calcul de visibilité, une telle représentation peut être obtenue en ajoutant des hyperplans au polytope, en prenant garde à ne pas modifier cette intersection. Cette technique est notamment utilisée par Bittner [Bittner 2002] et Nirenstein [Nirenstein 2003] (cf. §2.4.1). Cela leur permet d'une part d'effectuer des calculs

par rapport aux sommets de la V -représentation, et d'autre part de limiter les calculs à une zone restreinte autour de la quadrique Ω .

2.4 L'occultation dans l'espace de Plücker

Les calculs d'occultation dans l'espace de Plücker tentent généralement d'étendre les méthodes fonctionnant en dimension 2. Malheureusement, les grandes différences entre un espace de droites de dimension 2 et l'espace de Plücker rendent de telles extensions bien difficiles, par manque d'une vision claire de l'espace de droites. Nous décrivons ici les méthodes dites d'occultation, consistant à calculer exactement l'ensemble de droites traversant une région de l'espace, et soit ce qui est vu dans un demi-espace soit l'ensemble de droites traversant une seconde région de l'espace. Puis nous discutons de la nécessité d'un espace de droites différent.

2.4.1 Les méthodes de calcul d'occultation dans \mathbb{P}^5

Le calcul de l'occultation consiste à retirer d'un polytope $\mathcal{P}_{\mathcal{S}}$, représentant les droites $\bigcap_{i=1}^s \text{stab}(\mathbf{S}_i)$, les parties représentant les droites $\bigcup_{j=1}^o \text{stab}(\mathbf{O}_j)$ passant par chaque occulteur. En dimension 3, cela revient à des calculs de type CSG dans \mathbb{P}^5 [Nirenstein 2002] : pour un occulteur \mathbf{O} , le polytope $\mathcal{P}_{\mathcal{S}}$ est découpé par les hyperplans associés aux arêtes de \mathbf{O} . Le résultat est un nouvel ensemble de polytopes, dont l'un représente les droites passant par les faces $(\mathbf{S}_i)_{i=1\dots s}$ et l'occulteur \mathbf{O} . Ce polytope peut être retiré de l'ensemble, ou bien marqué comme représentant des droites occultées par \mathbf{O} .

Effectué sur chaque occulteur, ce calcul résulte en une chaîne de polytopes représentant chacun une partie des droites $\bigcap_{i=1}^s \text{stab}(\mathbf{S}_i) \setminus \bigcup_{j=1}^o \text{stab}(\mathbf{O}_j)$ passant par tous les $(\mathbf{S}_i)_{i=1\dots s}$, mais pas par les occulseurs $(\mathbf{O}_j)_{j=1\dots o}$.

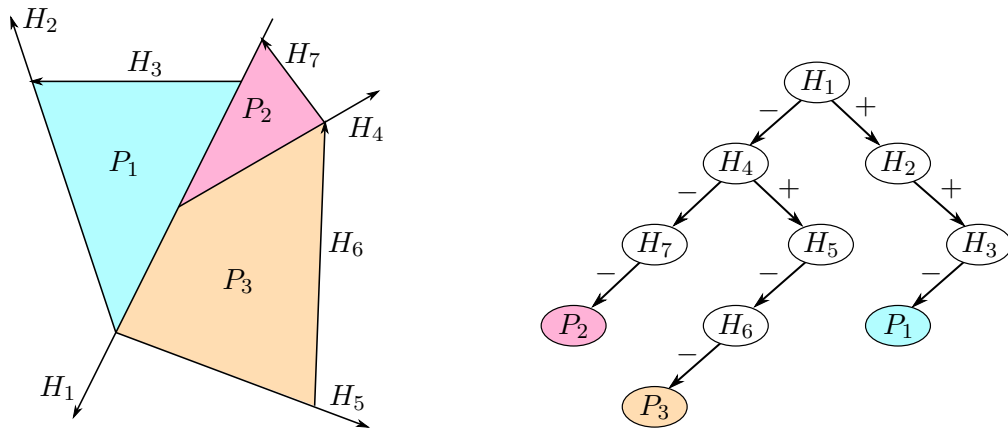
Les principales différences entre les différentes méthodes de calcul d'occultation concernent :

- La structure de données permettant de représenter les polytopes.
- Les algorithmes de découpe d'un polytope par un hyperplan.
- L'information du résultat final.

Les paragraphes suivants détaillent ces différences.

2.4.1.1 Arbres PSP de Pu [Pu 1999, Pu 1998]

Pu [Pu 1998, Pu 1999] propose un calcul de la visibilité depuis un polygone source (*from-polygon visibility*) en construisant un arbre PSP (*Plücker Space Partition*, un arbre BSP dans l'espace de Plücker). Un tel arbre représente des polytopes implicitement, en subdivisant \mathbb{P}^5 par des hyperplans selon le principe de Naylor [Naylor 1990b] : chaque nœud interne de l'arbre contient un hyperplan subdivisant l'espace en deux demi-espaces et chaque feuille représente un polytope (*cf.* figure 2.8).



(a) Trois polygones P_1 , P_2 et P_3 dans le plan, caractérisés par une partition binaire du plan par des hyperplans. L'orientation des hyperplans détermine le demi-espace positif et le demi-espace négatif.

(b) Arbre binaire représentant les trois polygones P_1 , P_2 et P_3 à partir d'une partition du plan. Les nœuds internes contiennent les hyperplans de la partition de l'espace, tandis que les feuilles représentent les polygones.

FIGURE 2.8 – Caractérisation d'une chaîne de polygones par une partition binaire du plan, représentée par un arbre BSP.

L'algorithme de Pu démarre avec un arbre représentant toutes les droites passant par un polygone source. Puis il traite les hyperplans correspondant aux arêtes des occulteurs : chaque hyperplan est inséré dans *toutes* les feuilles de l'arbre. Le résultat est l'arrangement d'hyperplans de Pellegrini [Pellegrini 1990, Pellegrini 1993], représentant la visibilité depuis le polygone source.

L'inconvénient de cette méthode est qu'elle ne tient pas compte de la localité de l'occultation due à un occulteur : les hyperplans coupent tout l'espace, y compris les régions non concernées par l'occulteur. Pu observe que sa méthode est soumise à la complexité théorique maximale d'un arrangement d'hyperplans dans \mathbb{P}^5 , soit $\mathcal{O}(n^4 \log n)$ [Aronov 1993]. L'explosion combinatoire due à un tel arrangement limite cette méthode à une quinzaine d'occulteurs. Les méthodes [Nirenstein 2003, Bittner 2002, Mora 2006] qui ont suivi ont notamment eu pour objectif de réduire cette complexité maximale, en effectuant des tests de rejet permettant de discriminer certaines découpes inutiles.

2.4.1.2 Arbre d'occultation de Bittner [Bittner 2002]

Sur le même principe d'arbre BSP dans l'espace des droites \mathbb{P}^5 , Bittner [Bittner 2002] propose une structure d'*arbre d'occultation* (*occlusion tree*) pour calculer la visibilité depuis un polygone. Si sa structure est sensiblement identique à celle de Pu, son calcul diffère sur plusieurs points.

L'algorithme de Bittner commence également avec un arbre représentant les droites pas-

sant par un polygone source. Les occulteurs sont ensuite considérés un à un, dans l'ordre de leur éloignement relatif par rapport à la source. Pour chaque occulteur, les représentations par sommets et par hyperplans du polytope associé aux droites passant par la source et l'occulteur sont calculées. Ces représentations sont déterminées à partir des hyperplans associés aux arêtes de la source et de l'occulteur. Comme ces hyperplans ne suffisent pas à déterminer une V -représentation (cf. le paragraphe 2.3.2.4), Bittner ajoute deux hyperplans supplémentaires à la H -représentation.

Ces représentations sont ensuite insérées dans l'arbre d'occultation. Bittner propose deux algorithmes :

1. Le premier est inspiré de [Naylor 1990a] ; il consiste à découper le polytope inséré par les hyperplans contenus dans les nœuds de l'arbre, puis à insérer récursivement les fragments obtenus dans les sous-arbres correspondants ; le fragment du côté positif (resp. négatif) de l'hyperplan dans le sous-arbre positif (resp. négatif).
2. Le second algorithme consiste à rejeter la V -représentation du polytope par ces mêmes hyperplans, *i.e.* à tester les sommets du polytope à partir de l'opérateur *side*, sans procéder à la découpe de celui-ci. Cela revient à déterminer de quel côté d'un hyperplan sont les sommets du polytope (cf. figure 2.9(b)). Le polytope est alors inséré globalement dans les sous-arbres où il n'est pas complètement rejeté.

Pour ces deux algorithmes, lorsqu'un polytope arrive dans une feuille, sa H -représentation la remplace. Ces deux algorithmes permettent de restreindre l'insertion des occulteurs dans la zone qu'ils occultent, réduisant ainsi la taille de l'arbre.

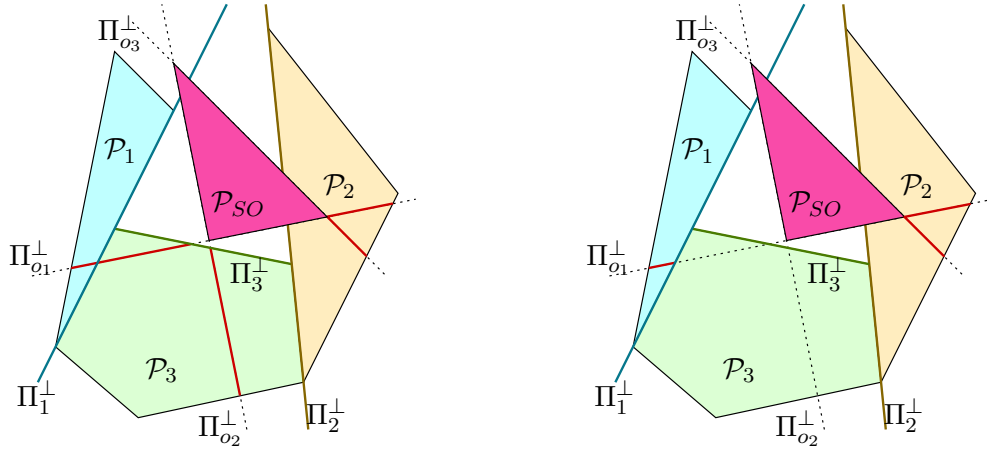
L'inconvénient du premier algorithme est l'évaluation, à chaque coupe du polytope, de la V -représentation des polytopes à partir de leur H -représentation. Cela nécessite de nombreux calculs de résolution de systèmes linéaires, particulièrement sensibles aux imprécisions numériques. Le second algorithme ne nécessite pas ces calculs et est donc plus rapide. Il est en revanche moins précis que le premier, vu que le polytope n'est pas découpé et que les rejets s'effectuent alors sur un polytope plus grand que nécessaire. Par conséquent, il arrive souvent qu'un hyperplan ne puisse pas rejeter un polytope non découpé, alors qu'il l'aurait fait sur un polytope découpé au fur et à mesure de son insertion dans l'arbre.

En conclusion, la méthode de Bittner vient avec deux algorithmes de construction, l'un imprécis et générant trop d'erreurs, et l'autre engendrant trop de calculs et donc limité quant au nombre d'occulteurs utilisables.

2.4.1.3 Visibilité entre deux polygones [Nirenstein 2003]

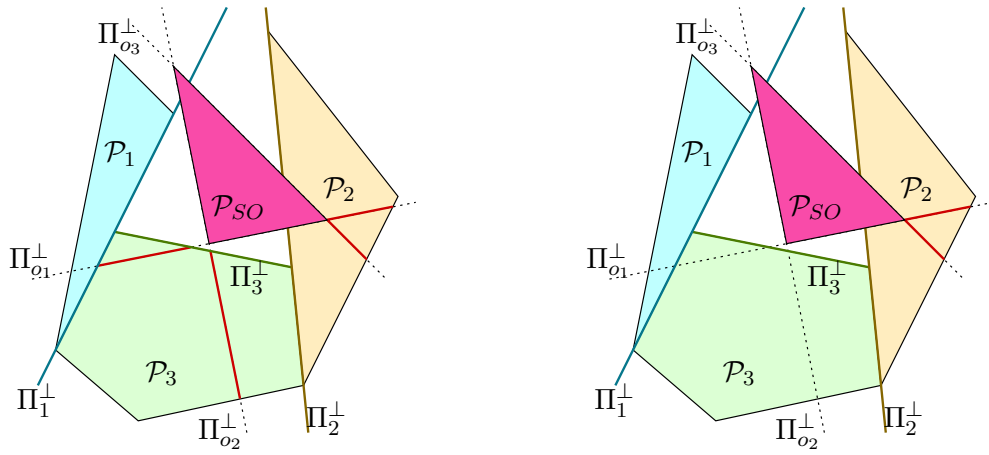
Parallèlement aux travaux de Bittner, Nirenstein définit un calcul de visibilité entre un polygone source S et un polygone cible C (*polygon-to-polygon visibility*). Le but est de calculer un prédicat de visibilité, *i.e.* obtenir une preuve de l'existence de visibilité entre deux polygones.

A la différence de Pu et de Bittner, Nirenstein n'utilise aucune structure d'arbre : un polytope est explicitement représenté par la topologie de son bord, en utilisant un graphe d'in-



(a) Découpe des polytopes dans \mathbb{P}^5 , sans aucun test de rejet.

(b) Rejet du polytope \mathcal{P}_{SO} par les nœuds de l'arbre BSP (cf. [Bittner 2002]).



(c) Rejet des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 par les hyperplans de l'occulteur \mathcal{P}_{SO} .

(d) Rejet de Mora : utilisation conjointe des tests de Bittner et Nirenstein.

FIGURE 2.9 – Test de rejet : (a) un arrangement d'hyperplans n'engendre aucun rejet, et donc une explosion du nombre de polytopes ; (b) rejet de Bittner, des occulteurs par les hyperplans Π_1^\perp , Π_2^\perp et Π_3^\perp , de l'arbre d'occultation ; ici le rejet permet d'éviter la découpe de \mathcal{P}_3 , mais pas de \mathcal{P}_1 ; (c) le test de Nirenstein permet d'éviter la découpe inutile de \mathcal{P}_1 , mais pas celle de \mathcal{P}_3 , en testant les sommets des polytopes à découper par les hyperplans de découpe ; (d) notre test : l'utilisation conjointe d'un arbre BSP et d'un graphe d'incidence permettant d'utiliser les tests de Bittner et de Nirenstein, réduisant considérablement le nombre de découpes inutiles.

cidence. Les nœuds du graphe représentent les sommets du polytope, et sont associés à une droite dans \mathbb{P}^5 . La découpe des polytopes par les hyperplans utilise une version optimisée

de l'algorithme de Bajaj et Pascucci [Bajaj 1996, Nirenstein 2003], permettant de traiter un complexe de polytopes en toute dimension.

Construction du polytope initial L'algorithme de Nirenstein commence par construire un polytope \mathcal{P}_{SC} , représentant les droites $stab(SC)$. Comme pour Bittner, ce polytope est au départ sous-contraint. Pour calculer ce polytope, Nirenstein part de toutes les droites sommet de S à sommet de C . Pour fermer ce polytope, Nirenstein détermine via une heuristique un couple de vecteurs \vec{r} et $-\vec{r}$ dans \mathbb{P}^5 . Ces deux vecteurs sont ensuite appliqués sur chaque sommet du polytope, afin d'engendrer un polytope plus grand mais fermé.

L'intérêt du vecteur \vec{r} est double selon Nirenstein. D'une part, il permet de déterminer un graphe d'incidence pour le polytope initial. D'autre part, il permet de réduire ce polytope à une zone proche de la quadrique \mathcal{Q} et limite ainsi les coupes inutiles des polytopes, apparaissant lorsqu'un polytope résultant ne coupe pas la quadrique.

Ordonnement des occulteurs Nirenstein définit également deux techniques améliorant considérablement la méthode de Pu, en réduisant la découpe de la visibilité. La première technique consiste à ordonner les occulteurs du plus important au moins important, en terme d'occultation, à partir d'une heuristique sur leurs tailles. Ces occulteurs sont ensuite utilisés dans l'ordre déterminé pour découper le polytope initial. L'idée est de retirer en premier les plus gros ensembles de droites occultées avec un nombre réduit de coupes.

Rejet des polytopes La seconde technique utilisée par Nirenstein [Nirenstein 2003] est un test consistant à rejeter les sommets d'un polytope contre les hyperplans associés à un occulteur (cf. figure 2.9(c)). Si tous les sommets d'un polytope sont du côté négatif d'un des hyperplans, alors on est assuré qu'aucune droite du polytope n'est occultée par cet occulteur. Dans ce cas, le polytope est simplement rejeté de la coupe.

En pratique, ce test de rejet réduit considérablement la complexité de la coupe des polytopes, comparée aux premiers travaux de Pu. Associée à l'ordonnement des occulteurs, cette technique est la principale source d'efficacité de la méthode de Nirenstein.

Silhouettes et représentation alternative des polytopes Haumont *et al.* [Haumont 2005] améliorent les travaux de Nirenstein en proposant un traitement de la silhouette et une représentation des polytopes à partir de leurs squelettes plutôt que de leurs graphes d'incidences [Haumont 2005].

La représentation d'un polytope par un squelette consiste en un graphe ne décrivant que les adjacences entre les sommets, les arêtes et les hyperplans : les nœuds du graphe sont les sommets du polytope tandis que les arcs sont ses arêtes. Les hyperplans sont connus en chaque nœud et arête par une liste, et servent à mettre à jour le polytope à chaque découpe.

2.4.2 Limites de ces méthodes

Reposant sur la paramétrisation de Plücker, les méthodes de calcul d’occultation offrent les solutions algorithmiques les plus efficaces pour déterminer puis appliquer une information de visibilité globale. Néanmoins, elles possèdent également deux limites principales.

2.4.2.1 Le problème du polytope minimal

Le problème de polytope minimal concerne la caractérisation du polytope initial représentant les droites passant par deux polygones convexes ; il apparaît avec les travaux de Nirenstein. Pour deux polygones S et C , le polytope minimal est le plus petit polytope dans \mathbb{P}^5 contenant l’ensemble $stab(SC)$ des droites passant par les deux polygones S et C .

Le polytope initial est sous-contraint. Bittner [Bittner 2002], Nirenstein [Nirenstein 2003], Haumont [Haumont 2005] et nous-même [Mora 2006] sommes contraints d’ajouter deux hyperplans, permettant de déterminer une V -représentation du polytope.

Nirenstein est le premier à souligner que plus les hyperplans supplémentaires sont «proches» de la quadrique Ω , et moins on aura de découpes inutiles, résultant en des polytopes ne représentant aucune droite. De la même façon, plus le polytope sera réduit et plus les tests de rejet seront précis et limiteront les découpes.

Nous donnerons la solution à ce problème dans le chapitre suivant, et montrerons qu’il n’est en réalité pas possible de déterminer cette solution à partir de deux hyperplans seulement.

2.4.2.2 Traitement des dégénérescences

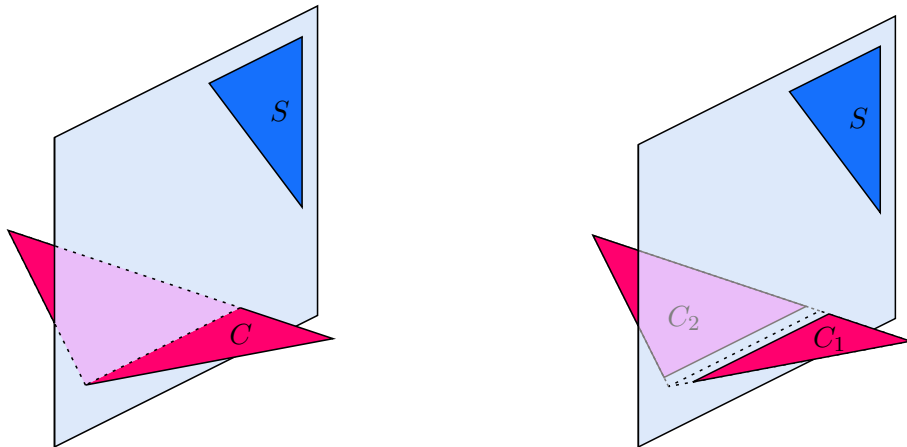
Dans le calcul de visibilité entre deux polygones S et C , une dégénérescence apparaît lorsque le plan support de la face S (respectivement de la face C) coupe le bord ou l’intérieur de la face C (respectivement la face S). La figure 2.10(a) montre un exemple de dégénérescence où le plan de la face S coupe l’intérieur de la face C .

Bien que ces cas ne soient pas clairement explicités dans les travaux précédents de Nirenstein, Haumont et Bittner, ils ne peuvent être traités de manière exacte dans ces méthodes. Pour transformer le cas dégénéré en un cas non dégénéré où le polytope est calculable, les faces coupées par le plan support de l’autre face sont “rognées” dans le voisinage de l’intersection (cf. figure 2.10(b)). Immédiatement, cette solution au problème est inexacte, car elle ne permet pas de représenter la visibilité dans la région supprimée. Ce traitement rend les méthodes précédentes inexactes, contrairement à ce qu’en disent leurs auteurs.

La nécessité de perturber la face C sur la figure 2.10(b), provient d’une projection des coordonnées de Plücker sur un hyperplan. Par exemple, pour une droite de coordonnées $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5)$, la projection sur la coordonnée π_0 donne :

$$(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5) \mapsto \left(1, \frac{\pi_1}{\pi_0}, \frac{\pi_2}{\pi_0}, \frac{\pi_3}{\pi_0}, \frac{\pi_4}{\pi_0}, \frac{\pi_5}{\pi_0}\right)$$

Cette projection revient à considérer les coordonnées homogènes normalisées d’un point de



(a) Dégénérescence : le plan support du polygone S coupe l'intérieur du polygone C.

(b) Rognage de la face C afin d'obtenir deux cas non dégénérés.

FIGURE 2.10 – Traitement des cas dégénérés dans les méthodes de Nirenstein [Nirenstein 2003], Haumont [Haumont 2005], et Mora [Mora 2006]. Exemple d'une dégénérescence pour le calcul de visibilité entre deux polygones S et C de l'espace tridimensionnel. La face C contient une intersection non vide avec le plan support de la face S, ce qui fait apparaître une dégénérescence.

l'espace arguésien (un espace affine fermé par l'infini), obtenu par *restriction* de l'espace \mathbb{P}^5 . Elle est utile dans ces travaux afin de simplifier l'élaboration des opérations géométriques dans \mathbb{P}^5 : la construction du polytope initial et le calcul d'intersection entre un polytope et un hyperplan dépendent de cette projection. Toutefois, les dégénérescences font apparaître des singularités, *i.e.* des points de \mathbb{P}^5 dont les coordonnées ne peuvent être normalisées (cas où $\pi_0 = 0$ dans l'exemple précédent), pour lesquelles les opérations géométriques ne sont pas définies. Cela explique pourquoi ces méthodes requièrent de transformer, par perturbation des faces, les cas dégénérés en cas non dégénérés, où de telles singularités n'apparaissent pas.

On remarque en outre que cette solution risque d'augmenter l'instabilité numérique du calcul. En effet, les points singuliers se transforment, par perturbation des faces, en points proches de points singuliers (π_0 tend vers zéro). Ces points sont normalisables, mais sont représentés par de grandes coordonnées, rendant la normalisation numériquement instable.

2.5 Conclusions

Ce chapitre a proposé un bref état de l'art des méthodes de calcul de visibilité globale, en insistant plus particulièrement sur les calculs d'occultation. Il détaille les calculs en dimension 2, qui sont notamment utilisés dans la thèse de P. Combeau (*cf.* chapitre 5). Il présente

aussi les techniques de calcul d'occultation en dimension 3 : bien qu'il s'agisse d'une extension de la méthode fonctionnant en dimension 2, nous avons montré qu'il existe beaucoup de différences dues au changement d'espace de droites. La méconnaissance de ce dernier, ainsi que sa dimension (il faut six coordonnées de Plücker pour discriminer une droite) rendent les méthodes précédentes difficiles à maîtriser et à comprendre.

Les problèmes pour un calcul d'occultation en dimension 3 sont nombreux :

- le calcul du polytope minimal qui n'est qu'approché dans les travaux précédents (et que nous avons corrigé avec F. Mora),
- l'explosion combinatoire qui limite le nombre d'occulteurs utilisables (que nous avons repoussé avec F. Mora)
- des dégénérescences qui rendent les méthodes inexactes (rognage des faces),
- des problèmes d'imprécision numérique, engendrés par une mauvaise interprétation de la notion d'espace "projectif",
- et la difficulté d'unifier les travaux en toute dimension (dans le but d'utiliser une unique méthode quelle que soit la dimension de l'espace géométrique) ...

Tous ces éléments sont, d'après nous, engendrés par la difficulté de penser dans un espace de droites en dimension 3. Le chapitre suivant présente nos travaux sur la visibilité : les premiers étendent ceux décrits dans ce chapitre, en repoussant les limites ; par la suite, la formalisation d'un espace de droites en dimension n nous permet d'unifier les méthodes en dimension quelconque, de résoudre le problème du polytope minimal, et de définir de nouvelles méthodes de calcul.

Contributions à la visibilité en dimension $n \geq 2$

Sommaire

3.1 Premiers travaux sur la visibilité	38
3.1.1 Calcul de visibilité en dimension 2	38
3.1.2 Calcul d’occultation en dimension 3	39
3.2 Introduction à l’algèbre de Grassmann	40
3.2.1 Présentation de l’algèbre de Grassmann	40
3.2.2 Propriétés et définitions utiles pour le calcul de visibilité	43
3.2.3 Algèbre de Grassmann sur un espace projectif \mathbb{P}^n	45
3.3 L’espace de droites en dimension n	48
3.3.1 L’espace \mathcal{D}_n et ses propriétés utiles	48
3.3.2 Le polytope minimal	53
3.4 Apports algorithmiques	54
3.4.1 Calcul d’occultation dans \mathcal{D}_3	55
3.4.2 Calcul à la demande	58
3.5 Conclusion	62

Le chapitre 2 a introduit les travaux précédents sur le calcul de visibilité en dimensions 2 et 3. Il a notamment insisté sur différents problèmes : pas de cohérence lors du changement de dimension empêchant la réutilisation des algorithmes existants ; un espace de droites aux propriétés incertaines qui nuit à la résolution de problèmes tels que les dégénérescences, à la classification de n’importe quel couple de droites, *etc.* ; le calcul du polytope minimal, *i.e.* l’ensemble des droites traversant deux ou plusieurs convexes dans l’espace géométrique, et qui est généralement défini à une heuristique près ...

Ce chapitre introduit nos propres travaux menés ces treize dernières années : d’abord en dimension 2 dans le cadre de la thèse de Pierre Combeau [Combeau 2004b] (encadrée avec Rodolphe Vauzelle) ; puis en dimension 3, dans le cadre des thèses de Frédéric Mora [Mora 2006] (encadrée avec Michel Mériaux) et de Sylvain Charneau [Charneau 2007b] (encadrée avec Laurent Fuchs et Pascal Lienhardt). Après quelques tentatives d’améliorations de l’existant (*cf.* section 3.1), nous avons étudié l’algèbre de Grassmann¹, et pu ainsi redéfinir formellement l’espace des droites en dimension quelconque et résoudre le problème du polytope minimal (*cf.* section 3.2). Ces travaux nous ont permis de mettre en

1. Dans la majorité des cas, nous ne rappelons pas dans ce mémoire les preuves des propriétés ou propositions ou théorèmes proposés ; elles sont accessibles dans la bibliographie.

place un calcul d'occultation bien plus performant que les précédents, tant en terme d'exactitude (sans approximation dans les cas dégénérés, qui de fait disparaissent) que de qualité (prise en compte d'une plus grande complexité visuelle). Ces apports algorithmiques sont détaillés dans la section 3.4. Au delà d'une méthode robuste de calcul d'occultation, nous détaillons notamment une méthode originale de calcul de visibilité *à la demande*, beaucoup plus légère en terme de mémoire consommée, et qui évite tout pré-calcul. Cette dernière fût développée en collaboration avec Frédéric Mora du département DMI de l'institut XLIM, Université de Limoges.

Dans la suite de ce chapitre, nous travaillons sur un espace vectoriel *réel* de dimension finie n , noté \mathbb{R}^n ou plus rarement E [Bass 1972].

3.1 Premiers travaux sur la visibilité

Nous résumons ici nos premiers travaux en dimension 2 et 3.

3.1.1 Calcul de visibilité en dimension 2

Dans le cadre de la thèse de P. Combeau [Combeau 2004b] et d'un contrat de recherche (CRE Topase, avec France-Télécom Recherche & Développement [Aveneau 2002]), nous avons utilisé les calculs d'occultation en dimension 2, afin d'accélérer les simulations radioélectriques (nous reviendrons en détails sur les aspects simulations dans le chapitre 5).

Afin d'obtenir le plus rapidement possible une caractérisation du canal, notamment pour y calculer des statistiques de qualité de réception en environnement urbain, nous avons cherché à calculer efficacement la carte de propagation d'une onde radio en dimension 2. La propagation part d'un émetteur, et s'effectue dans le domaine continu : l'objectif est de partitionner le plan en zones, chacune caractérisée par la même suite des mêmes phénomènes électromagnétiques (par exemple : une réception directe + une réflexion + une diffraction puis une réflexion). Dans chacune de ses zones, nous calculons une moyenne du signal reçu en 2 récepteurs dont les positions sont choisies de façon empirique.

Nous avons dans ce cadre utilisé une notion de faisceaux de droites : un point P de l'espace géométrique est associé à une droite dans l'espace de droites, correspondant à toutes les droites passant par P ; ainsi, un faisceau n'est qu'un sous-ensemble continu des droites passant par le sommet du faisceau ; dans l'espace des droites, un tel faisceau est donc un segment de droite. Par conséquent, les calculs de propagation s'effectuent en partie dans l'espace de droites : partant d'un faisceau théorique (par exemple l'un des quatre faisceaux issus d'un émetteur ponctuel, définis avec les quatre bords du rectangle englobant de la scène), nous cherchons à calculer le plus rapidement possible comment il doit être découpé en tenant compte de la géométrie de la scène.

En raison du phénomène de réflexion, un faisceau est très souvent tronqué : à sa base après une réflexion sur un mur, ou bien à son extrémité par les bords de la scène ou justement par un mur. Afin de calculer les limites d'un faisceau, il est nécessaire de chercher la géométrie

contenue dans le faisceau qui soit la plus proche de la base du faisceau. Dans l'espace des droites, cela revient à chercher les intersections d'un segment de droite (correspondant au faisceau) avec des droites (correspondant aux sommets de la scène).

Afin d'obtenir un calcul le plus efficace possible, nous avons combiné les calculs entre l'espace géométrique et l'espace de droites. Avec des techniques issues de la géométrie discrète, nous avons optimisé la recherche des sommets susceptibles de découper un faisceau donné.

Ces travaux ont été publiés dans [Combeau 2002b, Combeau 2002c, Aveneau 2003a, Combeau 2003, Combeau 2004a, Combeau 2006, Combeau 2007].

3.1.2 Calcul d'occultation en dimension 3

Dans le cadre de la thèse de F. Mora [Mora 2006], nous avons proposé d'utiliser les avantages des deux méthodes de Nirenstein et Bittner afin de décrire la visibilité globale entre deux polygones convexes [Mora 2005c] : nous utilisons un arbre BSP (similaire à celui de Bittner) afin de permettre ensuite l'extraction de l'information de visibilité ; nous associons cet arbre à la représentation explicite des polytopes pour calculer les découpes (Nirenstein). Cette information est ensuite utilisée en rendu d'ombres douces [Mora 2005a], puis en simulation de propagation d'ondes radio [Mora 2005b]. Notons que notre arbre BSP n'est pas un arbre d'occultation au sens de Bittner : les occulteurs ne sont pas insérés par ordre de profondeur depuis la source, ce qui évite les erreurs liées à ce genre d'ordonnement.

Cet arbre nous permet dans un premier temps d'annuler la découpe inutile d'un polytope, y compris lorsqu'elle est détectée tardivement. En effet, découper un polytope passe par des calculs numériques, dont les imprécisions intrinsèques rendent les données de plus en plus approchées. Or, les tests de rejet ou d'existence de droites dans un polytope sont particulièrement sensibles à la qualité des données manipulées. Nous avons remarqué que les découpes engendrent souvent des polytopes ne contenant plus aucune droite de l'espace \mathcal{A}^3 (leur intersection avec la Grassmannienne est vide). Annuler ces découpes inutiles permet alors de revenir à un polytope dont les sommets et les bords sont de meilleure qualité.

Nous avons également cherché à augmenter la complexité des scènes pour lesquelles la visibilité peut être calculée. Dans ce cadre, nous avons proposé d'utiliser la complémentarité de l'association des deux tests de rejet proposés par Bittner et par Nirenstein, permettant de réduire davantage la découpe de la visibilité. En pratique, cette complémentarité est rendue possible grâce à l'association de notre arbre BSP et de la représentation explicite des polytopes de \mathbb{P}^5 . Enfin, pour obtenir une méthode de rejet optimale, nous avons proposé une amélioration du test de Bittner, en utilisant le théorème suivant (rappelons que l'opérateur *side* \odot donne l'orientation de deux droites connues par leurs coordonnées de Plücker) :

Théorème 3.1 *Soit \mathbf{A} et \mathbf{B} deux polygones convexes ayant respectivement n et m sommets. Soit $vvset = \{v_{ij}, i \in [1, n], j \in [1, m]\}$ l'ensemble des droites v_{ij} passant par un sommet de \mathbf{A} puis un sommet de \mathbf{B} . Soit l une droite quelconque, et l' une droite traversant \mathbf{A} puis \mathbf{B} . Alors :*

$$\begin{aligned} \Pi_l \odot \Pi_x &\geq 0, \forall x \in vvset \rightarrow \Pi_l \odot \Pi_{l'} \geq 0, \\ \Pi_l \odot \Pi_x &\leq 0, \forall x \in vvset \rightarrow \Pi_l \odot \Pi_{l'} \leq 0. \end{aligned}$$

Une démonstration de ce théorème est disponible dans [Mora 2012]. Ce théorème nous permet de définir un polytope minimal sans utiliser de conjecture comme le font Bittner et Nirenstein. De plus, il permet d'éviter le rognage des faces comme Nirenstein : un produit nul est admis, l'opérateur *side* étant utilisée comme une relation d'ordre non strict. Cependant, il ne permet pas encore de corriger les cas dégénérés (ceux pour lesquels l'orientation des droites de *vset* n'est pas consistante, *i.e.* lorsque le plan support de la source coupe la cible, ou vice-versa ; *cf.* §2.4.2.2). Nous proposerons une version plus intéressante de ce théorème dans la section 3.3, caractérisant le polytope minimal en toutes circonstances.

Si ces améliorations ne nous ont pas permis d'utiliser des scènes quelconques, elles furent néanmoins nécessaires pour permettre une véritable exploitation de la visibilité entre deux polygones ; sans elles, l'extraction des ombres douces depuis l'information de visibilité ne fonctionne que très rarement (*cf.* section 4.1).

3.2 Introduction à l'algèbre de Grassmann

L'algèbre de Grassmann [Grassmann 1844] trouve ses racines au $XIX^{\text{ème}}$ siècle avec les travaux de Hermann Günther Grassmann (1809-1877). Son intérêt en géométrie provient de sa capacité à représenter des sous-espaces vectoriels par des termes algébriques [Parrochia 2012]. Elle fût ensuite étendue par les travaux de William K. Clifford à la fin du $XIX^{\text{ème}}$ siècle [Clifford 1876]. Longtemps ces travaux restèrent ignorés, au profit de ceux de Hamilton ou de Gibbs. Nous devons à David Hestenes, entre autres, leur réintroduction dans la communauté scientifique [Hestenes 1966, Hestenes 1984, Hestenes 1986, Hestenes 2001, Dorst 2007].

3.2.1 Présentation de l'algèbre de Grassmann

3.2.1.1 Le produit extérieur \wedge

Nous essayons ici d'éviter une présentation trop mathématique de l'algèbre de Grassmann. En 1843, W. H. Hamilton invente les quaternions, en se basant sur les travaux antérieurs d'Euler [Hamilton 1853]. Suivant un cheminement parallèle, en 1844 et en 1862 H. G. Grassmann propose un nouveau produit, qui peut de nos jours être vu comme une généralisation du produit vectoriel de la géométrie classique [Grassmann 1844, Grassmann 1862]. Rappelons que ce dernier, que nous noterons \times , calcule le vecteur v orthogonal à deux vecteurs a et b quelconques en dimension 3 ; le module de v correspond au produit des normes de a et de b multiplié par le sinus de l'angle entre a et b :

$$|v| = a \times b = |a| \cdot |b| \sin(\widehat{a, b}).$$

Si le produit vectoriel classique n'est défini qu'en dimension 3, le produit de Grassmann fonctionne en toute dimension. Nous donnons directement ses propriétés.

Propriété 3.1 Soit a, b et c trois vecteurs de l'espace vectoriel \mathbb{R}^n , et λ un réel de \mathbb{R} . Le produit de Grassmann ou produit extérieur, que nous noterons " \wedge " possède les propriétés suivantes.

Prop. 1 Il est anticommutatif :

$$a \wedge b = -b \wedge a.$$

Prop. 2 Il est bilinéaire :

$$\begin{aligned} (a + b) \wedge c &= a \wedge c + b \wedge c, \\ a \wedge (b + c) &= a \wedge b + a \wedge c, \\ (\lambda a) \wedge b &= a \wedge (\lambda b) = \lambda (a \wedge b). \end{aligned}$$

De la première propriété vient immédiatement le corollaire suivant, qui se révèle essentiel dans toute manipulation du produit extérieur.

Corollaire 3.1 Soit (a_i) pour $i \in [1 \dots k]$ une suite de vecteurs non nul de \mathbb{R}^n . Alors :

$$\begin{aligned} \forall i \in [1 \dots k], a_i \wedge a_i &= 0, \\ \{\exists i \neq j, a_i \sim a_j\} &\Leftrightarrow a_1 \wedge a_2 \wedge \dots \wedge a_k \neq 0. \end{aligned}$$

Nous avons notamment la règle suivante.

Lemme 3.1 Soient (a_1, \dots, a_p) une famille de vecteurs de \mathbb{R}^n et σ , une permutation de la suite $(1, \dots, p)$, de signature $\varepsilon_\sigma \in \{1, -1\}$. Par associativité du produit extérieur, il vient :

$$a_1 \wedge a_2 \wedge \dots \wedge a_p = \varepsilon_\sigma (a_{\sigma(1)} \wedge a_{\sigma(2)} \wedge \dots \wedge a_{\sigma(p)}) \quad (3.1)$$

Ces quelques règles de calculs très simples seront appliquées dans la suite de ce chapitre, souvent en ne rappelant d'ailleurs que la propriété d'anticommutativité. Il est donc conseillé au lecteur de bien se familiariser avec elles, par exemple en démontrant le corollaire 3.1 et le lemme 3.1.

3.2.1.2 L'algèbre de Grassmann

Le produit extérieur nous permet de construire directement l'algèbre de Grassmann. Nous en donnons la définition suivante :

Définition 3.1 L'algèbre de Grassmann, notée $\wedge(\mathbb{R}^n)$, est l'algèbre $(\mathbb{R}^n, +, \cdot, \wedge)$, dans laquelle :

- $(\mathbb{R}^n, +, \cdot)$ est un espace vectoriel de loi interne " $+$ " et de loi de composition externe " \cdot ".²
- \wedge est la loi de composition interne de $\wedge(\mathbb{R}^n) \times \wedge(\mathbb{R}^n)$ dans $\wedge(\mathbb{R}^n)$.

2. Bien entendu, cette loi de $\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ n'a rien à voir avec le produit scalaire.

A l'image de l'algèbre des quaternions, il s'agit évidemment d'une algèbre unifère³, associative mais non commutative. La propriété d'anticommutativité du produit extérieur permet de montrer facilement⁴ la proposition suivante.

Proposition 3.1 *L'algèbre de Grassmann $\bigwedge(\mathbb{R}^n)$ est de dimension 2^n .*

3.2.1.3 L'espace des multivecteurs

L'algèbre de Grassmann est un ensemble de 2^n sous-espaces vectoriels, chacun étant caractérisé par son *grade*. Avant de définir le grade d'un élément de $\bigwedge(\mathbb{R}^n)$, nous définissons la notion de multivecteur.

Définition 3.2 *Les éléments de $\bigwedge(\mathbb{R}^n)$ sont appelés multivecteurs.*

Tout multivecteur est une combinaison linéaire de produits de vecteurs d'une base de l'espace vectoriel \mathbb{R}^n . Nous avons alors la propriété et les définitions suivantes.

Propriété 3.2 *Soit (e_1, \dots, e_n) une base de \mathbb{R}^n . Les multivecteurs $e_{i_1} \wedge e_{i_2} \wedge \dots \wedge e_{i_k}$, pour toute suite $(i_j)_{j=1..k}$, $1 \leq i_1 < i_2 < \dots < i_k \leq n$ et $k \geq 0$, forment une base de l'algèbre $\bigwedge(\mathbb{R}^n)$.*

Définition 3.3 *Un multivecteur de $\bigwedge(\mathbb{R}^n)$ est homogène d'ordre k , s'il est combinaison linéaire de produits de k vecteurs de \mathbb{R}^n . Un tel multivecteur est appelé k -vecteur, et k est son grade.*

Par convention, un vecteur de \mathbb{R}^n est un 1-vecteur (multivecteur d'ordre 1), un scalaire est un 0-vecteur (multivecteur d'ordre 0).

Les propriétés 3.2 et d'anticommutativité ainsi que la définition 3.3 nous permettent d'énoncer le corollaire suivant.

Corollaire 3.2 *L'ensemble des multivecteurs homogènes d'ordre k est un sous-espace vectoriel de $\bigwedge(\mathbb{R}^n)$, noté $\bigwedge^k(\mathbb{R}^n)$, vérifiant les propriétés suivantes :*

$$\begin{aligned} \bigwedge^k(\mathbb{R}^n) &= \overbrace{\mathbb{R}^n \wedge \dots \wedge \mathbb{R}^n}^k \\ \bigwedge^k(\mathbb{R}^n) &= \{0\} \quad \text{pour tout } k > n \\ (\bigwedge^k(\mathbb{R}^n)) \wedge (\bigwedge^j(\mathbb{R}^n)) &= \bigwedge^{k+j}(\mathbb{R}^n) \\ \bigwedge(\mathbb{R}^n) &= \bigoplus_n \bigwedge^k(\mathbb{R}^n) \\ \dim \bigwedge^k(\mathbb{R}^n) &= C_n^k \\ \dim \bigwedge(\mathbb{R}^n) &= \sum_{k=0}^n C_n^k = 2^n \end{aligned}$$

3. Une algèbre est unifère lorsqu'elle admet un élément neutre 1 pour sa loi de composition ; dans notre cas nous pouvons écrire $1 \wedge m = m \wedge 1 = m$ pour tout élément m de $\bigwedge(\mathbb{R}^n)$.

4. C'est d'ailleurs un des exercices que je donne à nos étudiants du M1-Informatique, dans le cours *Informatique Graphique*.

Nous énonçons maintenant un corollaire et une définition, souvent utiles en pratique.

Corollaire 3.3 *Soit l'algèbre de Grassmann $\bigwedge(\mathbb{R}^n)$, sur l'espace vectoriel \mathbb{R}^n de dimension n . Pour tout k dans $[0; n]$, il existe un morphisme bijectif de $\bigwedge^k(\mathbb{R}^n)$ dans $\bigwedge^{n-k}(\mathbb{R}^n)$.*

Définition 3.4 *Soit l'algèbre de Grassmann $\bigwedge(\mathbb{R}^n)$. On appelle pseudoscalaire, que l'on note \mathfrak{I}_n , l'unique n -vecteur, à un facteur près non nul, générateur de l'espace $\bigwedge^n(\mathbb{R}^n)$ de dimension 1.*

La définition d'une bijection entre $\bigwedge^n(\mathbb{R}^n)$ et $\bigwedge^0(\mathbb{R}^n) \simeq \mathbb{R}$ revient à fixer un pseudoscalaire unitaire.

3.2.1.4 Exemple de l'algèbre $\bigwedge(\mathbb{R}^3)$

Prenons en exemple l'algèbre de Grassmann construite sur \mathbb{R}^3 , lui-même muni de la base (e_1, e_2, e_3) . Une base de $\bigwedge(\mathbb{R}^3)$ est la famille :

$$(1, e_1, e_2, e_3, e_1 \wedge e_2, e_1 \wedge e_3, e_2 \wedge e_3, e_1 \wedge e_2 \wedge e_3)$$

De plus, nous avons :

$$\begin{array}{ll} (1) & \text{est une base de } \bigwedge^0(\mathbb{R}^3) \\ (e_1, e_2, e_3) & \text{est une base de } \bigwedge^1(\mathbb{R}^3) \\ (e_1 \wedge e_2, e_1 \wedge e_3, e_2 \wedge e_3) & \text{est une base de } \bigwedge^2(\mathbb{R}^3) \\ (e_1 \wedge e_2 \wedge e_3) & \text{est une base de } \bigwedge^3(\mathbb{R}^3) \end{array}$$

Clairement, la dimension de $\bigwedge^0(\mathbb{R}^3)$ égale celle de $\bigwedge^3(\mathbb{R}^3)$, et celle de $\bigwedge^1(\mathbb{R}^3)$ vaut celle de $\bigwedge^2(\mathbb{R}^3)$: il est facile de définir une bijection entre ces espaces. De plus, $e_1 \wedge e_2 \wedge e_3$ est à lui seul une famille génératrice de $\bigwedge^3(\mathbb{R}^3)$: c'est le pseudoscalaire \mathfrak{I}_3 .

3.2.2 Propriétés et définitions utiles pour le calcul de visibilité

Nous complétons notre introduction de l'algèbre de Grassmann par quelques propriétés nécessaires à la suite de ce chapitre.

3.2.2.1 Décomposabilité des multivecteurs

Dans l'algèbre de Grassmann, la notion de divisibilité est la suivante.

Définition 3.5 *Un k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$ est divisible par le vecteur $v \in \mathbb{R}^n$ si $\exists u \in \bigwedge^{k-1}(\mathbb{R}^n)$ tel que $M = u \wedge v$.*

La définition 3.5 peut être interprétée comme une définition de divisibilité à droite, posant la question de la divisibilité à gauche. Néanmoins, si un vecteur $v \in \mathbb{R}^n$ divise le k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$, alors $M = u \wedge v$ est vrai pour au moins un vecteur $u \in \bigwedge^{k-1}(\mathbb{R}^n)$. L'anticommutativité du produit extérieur permet de conclure que M peut aussi s'exprimer comme $M = (-1)^{k-1} v \wedge u$. Ainsi, la condition "être divisible" ne dépend pas (à un signe près) de la position du vecteur v dans l'expression de M .

Définition 3.6 Un k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$ est dit totalement décomposable si il existe k vecteurs $v_1, \dots, v_k \in \mathbb{R}^n$ tels que $M = v_1 \wedge \dots \wedge v_k$. M est alors appelé une k -lame (ou k -blade en anglais).

Propriété 3.3 Un vecteur $v \in \mathbb{R}^n$ divise le k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$ si et seulement si $M \wedge v = 0$.

Démonstration

Lemme 1.6 dans [Kolhatkar 2004].

□

Propriété 3.4 Soit le k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$. L'ensemble $\{v \in \mathbb{R}^n \mid M \wedge v = 0\}$ est un sous-espace vectoriel de \mathbb{R}^n , que nous appelons le sous-espace des diviseurs de M .

Démonstration

Trivial en vertu de la linéarité à droite du produit extérieur.

□

Propriété 3.5 Le k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$ est dit totalement décomposable si et seulement si son sous-espace de diviseurs est de dimension k .

Démonstration

Théorème 1.1 dans la section 4.1 de [Marcus 1975].

□

Corollaire 3.4 Le k -vecteur $M \in \bigwedge^k(\mathbb{R}^n)$ est totalement décomposable si et seulement si le noyau de l'application linéaire $\phi_M : \mathbb{R}^n \rightarrow \bigwedge^{k+1}(\mathbb{R}^n) : v \rightarrow M \wedge v$ est de dimension k .

Démonstration

Le noyau $\text{Ker}(\phi_M)$ est égal au sous-espace des diviseurs de M .

□

Toutes les définitions et résultats proposés dans ce paragraphe peuvent trivialement être généralisés par linéarité au cas des multivecteurs, en prenant en compte que les multivecteurs sont des éléments de l'espace vectoriel gradué :

$$\bigwedge(\mathbb{R}^n) = \bigoplus_{k=0}^n \bigwedge^k(\mathbb{R}^n).$$

3.2.2.2 Notion de variété de Grassmann

L'ensemble des multivecteurs non nuls, décomposables d'ordre k , représentent donc dans $\bigwedge(\mathbb{R}^n)$ l'ensemble des sous-espaces vectoriels de dimension k dans \mathbb{R}^n , pour tout $k \in [0; n]$. Ces multivecteurs décomposables donnent une représentation algébrique des variétés de Grassmann (cf. §2.3.2.3).

Définition 3.7 *L'ensemble de tous les sous-espaces vectoriels de dimension k , $0 \leq k \leq n$, dans un espace vectoriel réel \mathbb{R}^n , décrit une variété projective généralement non linéaire de l'espace $\mathbb{P}(\bigwedge^k(\mathbb{R}^n))$, appelée variété de Grassmann ou grassmannienne et notée $G^{\mathbb{R}}(k, n)$. Les points de cette variété ont pour représentants homogènes les k -vecteurs décomposables non nuls de $\bigwedge^k(\mathbb{R}^n)$.*

Par exemple, les 1-vecteurs sont tous immédiatement décomposables. La grassmannienne $G^{\mathbb{R}}(1, n)$ correspond précisément à l'espace projectif \mathbb{P}^n . Comme nous l'avons vu dans le chapitre précédent, la grassmannienne joue un rôle primordial dans le calcul de la visibilité globale en dimension 3 (et au delà).

3.2.3 Algèbre de Grassmann sur un espace projectif \mathbb{P}^n

Nous introduisons ici la représentation des variétés linéaires projectives de \mathbb{P}^n grâce à l'algèbre de Grassmann. À cette fin, nous posons nos notations de l'espace géométrique en dimension n [Audin 2004], nous rappelons la notion de variété linéaire projective [Casanova 1965, Cagnac 1965], puis nous présentons le plongement des variétés linéaires projectives dans l'algèbre, et enfin nous traitons de leur orientation.

3.2.3.1 L'espace géométrique : notations et hypothèses

Soit l'espace projectif réel \mathbb{P}^n de dimension $n \geq 2$, déduit de l'espace \mathbb{R}^{n+1} dans lequel sont plongés les objets géométriques. Dans la suite de ce mémoire, nous notons cet espace \mathfrak{G}_n , en référence à l'espace *géométrique*, *i.e.* l'espace dont on étudie la géométrie.

Les objets considérés dans le calcul de visibilité globale dans \mathfrak{G}_n sont des polytopes convexes (des polygones convexes en dimension quelconque, voir définition 2.6, page 26), aussi appelés faces. Par exemple, une 0-face sera un sommet ; une 1-face, une arête ; une 2-face un polygone ; *etc.*

Par convention, les faces de \mathfrak{G}_n seront notées en police grasse, tandis que les variétés linéaires (*i.e.* les multivecteurs de $\bigwedge(\mathbb{R}^{n+1})$ représentant ces variétés) seront notées en police mathématique standard. Par exemple, la $(n - 1)$ -face \mathbf{F} a pour support un hyperplan (une $(n - 1)$ -variété) F .

3.2.3.2 Variétés linéaires projectives dans \mathfrak{G}_n

Une variété linéaire projective se définit ainsi.

Définition 3.8 Soit \mathbb{P}^n un espace projectif de dimension n , obtenu par les 1-sous-espaces de \mathbb{R}^{n+1} . Soit \mathbb{P}' une partie non vide de \mathbb{P}^n . L'ensemble \mathbb{P}' est un sous-espace projectif de \mathbb{P}^n si et seulement si l'ensemble \mathbb{R}' des vecteurs homogènes représentant tous les points de \mathbb{P}' est un sous-espace vectoriel de \mathbb{R}^{n+1} . \mathbb{P}' est alors appelé variété linéaire projective de \mathbb{P}^n .

Si \mathbb{R}' est de dimension k , alors la variété \mathbb{P}' est de dimension $k - 1$.

Ceci nous permet d'introduire la notion de points projectifs indépendants.

Définition 3.9 Soient \mathbb{P}^n , l'espace projectif réel de dimension n et P_1, \dots, P_k k points projectifs dans \mathbb{P}^n . Les points P_1, \dots, P_k sont projectivement indépendants si et seulement si leurs représentants homogènes p_1, \dots, p_k sont linéairement indépendants.

La proposition suivante rappelle que k points indépendants sont générateur d'une unique variété linéaire projective de dimension k .

Propriété 3.6 Par k points projectifs indépendants dans \mathbb{P}^n passe une et une seule k -variété linéaire projective. Il s'agit de la variété générée par le sous-espace de \mathbb{R}^{n+1} ayant une base formée par les représentants homogènes des k points. On dit alors que les k points génèrent la variété.

3.2.3.3 Variétés linéaires projectives de \mathfrak{G}_n dans $\bigwedge(\mathbb{R}^{n+1})$

De manière générale, une variété linéaire projective de dimension k est un sous-espace projectif de \mathfrak{G}_n , dont le générateur est un $(k + 1)$ -sous-espace vectoriel de l'espace \mathbb{R}^{n+1} . Par exemple, un point est une variété de dimension 0 dont le générateur est un 1-sous-espace de \mathbb{R}^{n+1} (cf. définition 3.8, page 46). Cette définition permet d'établir une correspondance biunivoque entre les variétés linéaires projectives de \mathfrak{G}_n et les sous-espaces vectoriels de \mathbb{R}^{n+1} .

Cette bijection est essentielle, puisque l'algèbre de Grassmann $\bigwedge(\mathbb{R}^{n+1})$ permet, fondamentalement, de représenter les sous-espaces vectoriels de \mathbb{R}^{n+1} . Plus précisément, un k -vecteur décomposable K , obtenu par le produit de k vecteurs de \mathbb{R}^{n+1} , représente un unique sous-espace vectoriel \mathcal{K} de dimension k , dans \mathbb{R}^{n+1} (cf. propriété 3.4). En revanche, les k -vecteurs non décomposables ne correspondent à aucun sous-espace vectoriel de \mathbb{R}^{n+1} . On a immédiatement la propriété classique suivante.

Propriété 3.7 Soient $k + 1$ points projectifs indépendants (P_0, \dots, P_k) de \mathfrak{G}_n , de représentants homogènes (p_0, \dots, p_k) dans \mathbb{R}^{n+1} . La k -variété linéaire projective, déterminée de manière unique par les points $(P_i)_{i=0 \dots k}$, a pour représentant homogène le $(k + 1)$ -vecteur $p_0 \wedge \dots \wedge p_k$ de $\bigwedge(\mathbb{R}^{n+1})$.

Ainsi, une variété linéaire de \mathfrak{G}_n est naturellement représentée par son ensemble de points générateurs. Par exemple, une droite de \mathfrak{G}_n , déterminée par deux points de représentants p_1 et p_2 dans \mathbb{R}^{n+1} , correspond dans $\bigwedge(\mathbb{R}^{n+1})$ au 2-vecteur $p_1 \wedge p_2$.

Cette propriété se déduit du fait que si k points (P_1, \dots, P_k) sont indépendants dans \mathfrak{G}_n , alors leurs représentants homogènes (p_1, \dots, p_k) le sont également dans \mathbb{R}^{n+1} . Le multivecteur $p_1 \wedge \dots \wedge p_k$ est donc non nul, et représente le sous-espace généré par les $p_i, i = 1 \dots k$. Il est le sous-espace, dans \mathbb{R}^{n+1} , générant la variété linéaire contenant les points P_1, \dots, P_k (cf. propriété 3.6).

3.2.3.4 Orientation des variétés linéaires projectives

La classification des droites de \mathfrak{G}_n , par rapport à des variétés linéaires, est un élément essentiel du calcul de visibilité dans l'espace des droites (cf. §2.2.1.4). Il repose sur la possibilité d'«orienter» les droites et plus généralement les variétés linéaires : la classification détermine l'orientation des droites relativement aux points en dimension 2 ou aux droites en dimension 3.

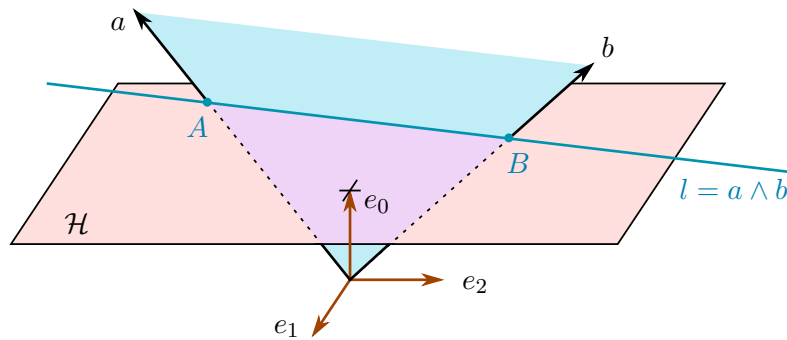


FIGURE 3.1 – Plan projectif \mathfrak{G}_2 , obtenu depuis \mathbb{R}^3 , de base (e_0, e_1, e_2) . La droite passant par les deux points A puis B est déterminée par le 2-sous-espace vectoriel orienté, représenté par le 2-vecteur décomposable $a \wedge b$, lui-même obtenu par le produit des représentants homogènes a et b des deux points A et B .

Dans l'algèbre de Grassmann, une variété linéaire est représentée par un multivecteur ; le signe du multivecteur caractérise l'orientation de la variété associée. Cette propriété est illustrée sur la figure 3.1⁵ : soient deux points projectifs A et B de \mathfrak{G}_2 , de représentants homogènes a et b dans \mathbb{R}^3 . La droite passant par A puis B est représentée par le 2-vecteur $a \wedge b$, tandis que la droite d'orientation opposée passant par B puis A est représentée par le multivecteur $b \wedge a = -a \wedge b$. Les deux multivecteurs $a \wedge b$ et $b \wedge a$ représentent le même sous-espace de \mathbb{R}^3 , donc la même variété de \mathfrak{G}_2 . Le changement de signe du multivecteur est alors interprété comme un changement d'orientation de la droite. Ceci se généralise à toute variété linéaire de \mathfrak{G}_n : le changement du signe du multivecteur représentant une variété est toujours interprété comme le changement d'orientation de la variété.

Cette orientation est *algébrique* : elle consiste à considérer un élément (un multivecteur)

5. Nous utilisons dans cette illustrations un hyperplan $\mathcal{H} = \{x \in \mathbb{R}^3 | x \cdot e_0 = 1\}$. Cela revient à interpréter l'espace projectif comme un espace Arguésien [Casanova 1965]. Cet hyperplan n'est utile que pour aider à l'interprétation des figures, en permettant de visualiser les vecteurs de \mathbb{R}^3 comme des *points* de l'espace.

et son opposé pour une loi de groupe (l'addition dans l'espace des multivecteurs), comme représentants des deux orientations d'une même variété. Cette orientation des variétés linéaires est équivalente à l'orientation algébrique de points projectifs.

Il est possible de représenter toute variété linéaire de \mathfrak{G}_n par un point projectif dont le représentant homogène est un multivecteur décomposable :

Propriété 3.8 Soit l'espace projectif $\mathbb{P}^{C_{n+1}^{k+1}-1} = \mathbb{P}(\wedge^{k+1}(\mathbb{R}^{n+1}))$, obtenu à partir des 1-sous-espaces de l'espace des $(k+1)$ -vecteurs de $\wedge(\mathbb{R}^{n+1})$. Chaque point projectif de $\mathbb{P}^{C_{n+1}^{k+1}-1}$, dont le représentant homogène est un $(k+1)$ -vecteur décomposable, correspond à une unique k -variété linéaire de \mathfrak{G}_n .

Cette correspondance est biunivoque, i.e. à toute k -variété linéaire projective de \mathfrak{G}_n correspond un unique point projectif dans $\mathbb{P}^{C_{n+1}^{k+1}-1}$.

L'ensemble de ces points décrit une variété généralement non linéaire dans $\mathbb{P}^{C_{n+1}^r-1}$, $r \in [0 \dots n+1]$, appelée *variété de Grassmann* ou *grassmannienne* et notée $G^{\mathbb{R}}(r, n+1)$. Il s'agit de manière équivalente de l'ensemble des r -sous-espaces vectoriels de \mathbb{R}^{n+1} (cf. définition 3.7). Au contraire, toujours dans cette interprétation des algèbres de Grassmann, les points projectifs dont les représentants homogènes ne sont pas décomposables n'appartiennent pas à la grassmannienne et ne représentent aucune variété linéaire de \mathfrak{G}_n .

Dans la suite de ce chapitre, la notion d'orientation des variétés linéaires fait toujours référence à la notion d'*orientation algébrique*.

3.3 L'espace de droites en dimension n

Le paragraphe 3.2.3.4 nous permet directement d'introduire l'ensemble des droites de \mathfrak{G}_n , comme étant la Grassmannienne $G^{\mathbb{R}}(1, n)$ ou encore $G^{\mathbb{R}}(2, n+1)$ en projectif. Cette dernière est donc une partie de $\wedge^2(\mathbb{R}^{n+1})$. Cette section définit plus formellement l'espace de droites associé à \mathfrak{G}_n , puis discute de l'orientation d'une droite avec une $(n-2)$ -face, et enfin donne l'exemple de \mathfrak{G}_3 .

3.3.1 L'espace \mathfrak{D}_n et ses propriétés utiles

Nous définissons ici l'espace de droites \mathfrak{D}_n , accompagné de propriétés importantes pour la visibilité.

3.3.1.1 L'espace \mathfrak{D}_n

Nous donnons la définition suivante d'un espace de droites associé à \mathfrak{G}_n .

Définition 3.10 L'espace des droites de \mathfrak{G}_n , noté \mathfrak{D}_n , est l'espace projectif $\mathbb{P}(\wedge^2(\mathbb{R}^{n+1}))$, déduit de l'espace des 2-vecteurs de l'algèbre $\wedge(\mathbb{R}^{n+1})$.

Il apparaît immédiatement que l'espace de droites contient plus que les droites de \mathfrak{G}_n , à part pour $n = 2$. En effet, les droites sont des points de la Grassmannienne, qui n'est pas une variété linéaire pour $n \geq 3$.

En plus de leurs qualités intrinsèques, les algèbres géométriques sont récréatives. Aussi, nous profitons de cette discussion sur la nature de l'espace de droites \mathfrak{D}_n pour montrer le lemme suivant.

Lemme 3.2 *Soit l'algèbre de Grassmann $\wedge(\mathbb{R}^3)$. Tout 2-vecteur non nul B de $\wedge^2(\mathbb{R}^3)$ est décomposable.*

Démonstration

Par l'absurde : soit (e_0, e_1, e_2) une base de \mathbb{R}^3 , et soit B le 2-vecteur non nul et non décomposable de coordonnées suivantes :

$$B = b_0 e_1 \wedge e_2 + b_1 e_0 \wedge e_2 + b_2 e_0 \wedge e_1,$$

Les trois coordonnées de B sont toutes non nulles : sinon B serait décomposable (soit $B = e_0 \wedge (b_1 e_2 + b_2 e_1)$, soit $B = e_1 \wedge (b_0 e_2 - b_2 e_0)$, soit $B = (b_0 e_1 + b_1 e_0) \wedge e_2$) ce qui est par hypothèse absurde.

Soit maintenant $a = (b_1 e_0 + b_0 e_1) \neq 0$. Alors :

$$\begin{aligned} B \wedge a &= (b_0 e_1 \wedge e_2 + b_1 e_0 \wedge e_2 + b_2 e_0 \wedge e_1) \wedge (b_1 e_0 + b_0 e_1), \\ &= b_0 b_1 e_1 \wedge e_2 \wedge e_0 + b_1 b_0 e_0 \wedge e_2 \wedge e_1 \end{aligned}$$

en vertu de l'anticommutativité du produit extérieur. Donc a divise B , puisque :

$$\begin{aligned} B \wedge a &= b_0 b_1 e_0 \wedge e_1 \wedge e_2 - b_1 b_0 e_0 \wedge e_1 \wedge e_2 \\ &= (b_0 b_1 - b_0 b_1) e_0 \wedge e_1 \wedge e_2 = 0. \end{aligned}$$

Soit le vecteur $c = b_2/b_1 e_1 + e_2$. Alors $a \wedge c = B$ et donc c divise B . Or, $a \wedge c = B \neq 0$ indique que a et c sont indépendants. Ainsi nous avons deux diviseurs indépendants, donc B est décomposable suivant la propriété 3.5. Ce qui est absurde, donc B et tous les 2-vecteurs sont décomposables.

□

Au delà de l'intérêt intrinsèque de ce lemme, sa démonstration illustre un indéniable avantage de l'utilisation de l'algèbre de Grassmann pour définir un espace de droite : il est très facile de manipuler ses termes pour effectuer des raisonnements et des démonstrations, et autorise l'utilisation d'un système de calcul formel.

3.3.1.2 Droites réelles et droites imaginaires

Mis à part le cas particulier de \mathfrak{G}_2 , l'espace de droites contient des droites dites réelles (ayant une correspondance dans \mathfrak{G}_n) et d'autres purement imaginaires (ne correspondant à aucune droite de \mathfrak{G}_n). Nous retrouvons là un fait connu dans le chapitre précédent en dimension 3. En utilisant la notion de décomposabilité, nous donnons cependant un mécanisme simple pour déterminer si un 2-vecteur est une droite réelle.

Propriété 3.9 Soit l'algèbre de Grassmann $\bigwedge(\mathbb{R}^{n+1})$, $n \geq 2$. Un 2-vecteur non nul B de $\bigwedge(\mathbb{R}^{n+1})$ est décomposable si et seulement si $B \wedge B = 0$.

Cette proposition permet de déterminer théoriquement si une droite de \mathfrak{G}_n est réelle. Nous savions que pour $n = 2$, le cas est trivial (toutes les droites sont réelles); pour $n = 3$ ce n'est plus vrai.

La question $B \wedge B = 0$ revient à calculer le 4-vecteur $B \wedge B$, i.e. un vecteur ayant par définition C_{n+1}^4 coordonnées. Cela revient à résoudre un système de C_{n+1}^4 équations $B_i \wedge B_j = 0$, encore une fois en vertu de l'anticommutativité du produit extérieur.

En dimension 3, l'espace $\bigwedge^4(\mathbb{R}^4)$ est de dimension C_4^4 , et donc une unique équation est à résoudre (cf. [Teller 1999]). En dimension 4 et au delà, la question de la nature d'une droite devient plus coûteuse en terme de calculs, même si l'utilisation d'un générateur de code [Fontijne 2003, Charneau 2005, Fontijne 2007] permet de limiter le temps de développement d'une solution algorithmique.

3.3.1.3 Orientation d'un droite et d'une $(n - 2)$ -face

Le chapitre 2 a montré que la clé d'un calcul de visibilité réside dans l'orientation, en dimension 2 entre une droite et un sommet, et en dimension 3 entre une droite et une autre droite. Cette notion se retrouve et surtout s'étend naturellement pour le calcul de visibilité en dimension n . En utilisant le produit extérieur de l'algèbre de Grassmann, la notion d'orientation est immédiate : elle correspond à celle de la variété construite à partir des deux k -vecteurs, ici entre une droite (un 2-vecteur) et une $(n - 2)$ -face (un $(n - 1)$ -vecteur).

Le théorème suivant montre que cette orientation fonctionne pour tout couple de vecteurs non nuls [Aveneau 2011].

Théorème 3.2 Soit δ un isomorphisme entre $\bigwedge^2(\mathbb{R}^{n+1})$ et $\bigwedge^{n-1}(\mathbb{R}^{n+1})$ ⁶. Alors, le produit interne entre tout couple de 2-vecteurs D_1 et D_2 est équivalent au calcul de l'orientation entre une droite et une $(n - 2)$ -face :

$$D_1 \cdot D_2 \equiv D_1 \wedge \delta(D_2) \equiv \delta(D_1) \wedge D_2,$$

à l'identification près des pseudoscalaires et des scalaires dans $\bigwedge(\mathbb{R}^{n+1})$.

La classification de deux 2-vecteurs consiste ainsi à considérer l'un des deux comme le dual d'un $(n - 1)$ -vecteur. La dualité étant toujours définissable de façon bijective à travers l'isomorphisme δ (cf. Corollaire 3.3 page 43), il n'y a aucun doute sur le fait que cela correspond bien au calcul de l'orientation entre une droite et une $(n - 2)$ -face (ou l'inverse).

La classification a les mêmes propriétés qu'un produit scalaire : contrairement aux travaux de Nirenstein [Nirenstein 2002], qui doit vérifier que les sommets du polytope représentant un ensemble de droites sont bien des droites réelles, avant de pouvoir effectuer une classification, nous pouvons manipuler n'importe quelle droite, réelle ou imaginaire.

6. Dans [Aveneau 2011] δ est définie en utilisant la contraction à gauche [Dorst 2007].

Le théorème 3.2 permet d'unifier les méthodes de calcul d'occultation en dimension $n \geq 2$. En effet, la classification fonctionne dans tout espace de droites \mathfrak{D}_n , et permet donc de définir l'ensemble des droites traversant une $(n - 1)$ -face convexe \mathbf{F} de l'espace comme l'ensemble des droites ayant une orientation cohérente par rapport aux bords de \mathbf{F} , *i.e.* des $(n - 2)$ -faces.

Théorème 3.3 *Soit \mathbf{F} une $(n - 1)$ -face convexe de \mathfrak{G}_n , supportée par une $(n - 1)$ -variété F et bordée par les $(n - 2)$ -faces convexes $(\mathbf{f}_i)_{i=1,\dots,r}$, elles-mêmes supportées par les $(n - 2)$ -variétés $(f_i)_{i=1,\dots,r}$.*

Il existe deux orientations possibles des $(n - 2)$ -variétés $(f_i)_{i=1,\dots,r}$, selon l'orientation algébrique de la variété F , telles qu'une droite D de \mathfrak{G}_n , non confondue avec l'hyperplan F , passe par la face \mathbf{F} si et seulement si elle vérifie :

$$D \wedge f_i \geq 0 \quad \text{pour tout } i = 1 \dots r, \quad (3.2)$$

$$\text{ou } D \wedge f_i \leq 0 \quad \text{pour tout } i = 1 \dots r. \quad (3.3)$$

L'ensemble des droites D de \mathfrak{G}_n passant par \mathbf{F} est l'intersection de la grassmannienne $G^{\mathbb{R}}(2, n+1)$ avec un polytope de \mathfrak{D}_n , déterminé par l'intersection des demi-espaces positifs associés aux hyperplans de \mathfrak{D}_n déduits des $(n - 2)$ -variétés f_i .

3.3.1.4 Nouvelle interprétation des droites de Plücker

L'espace de droites \mathfrak{D}_3 et l'espace de Plücker manipulent des droites de \mathfrak{G}_3 ; il est donc légitime d'essayer de les comparer. En pratique, les coordonnées de Plücker se retrouvent en tant que coordonnées d'un 2-vecteur.

Soit (e_0, e_1, e_2, e_3) une base de \mathfrak{G}_n , telle que e_0 soit associé à la coordonnée homogène. La droite passant par deux sommets de coordonnées respectives $(1, x_1, y_1, z_1)$ et $(1, x_2, y_2, z_2)$ est associée au 2-vecteur suivant :

$$\begin{aligned} \Pi &= (e_0 + x_1 e_1 + y_1 e_2 + z_1 e_3) \wedge (e_0 + x_2 e_1 + y_2 e_2 + z_2 e_3), \\ &= (x_2 - x_1) e_0 \wedge e_1 + (y_2 - y_1) e_0 \wedge e_2 + (z_2 - z_1) e_0 \wedge e_3 \\ &\quad + (y_1 z_2 - z_1 y_2) e_2 \wedge e_3 + (z_1 x_2 - x_1 z_2) e_3 \wedge e_1 + (x_1 y_2 - y_1 x_2) e_1 \wedge e_2. \end{aligned}$$

La définition 2.3 (*cf.* page 23) des coordonnées de Plücker permet de conclure qu'elles correspondent à celles d'un 2-vecteur de \mathfrak{G}_3 :

$$\Pi = \pi_0 e_0 \wedge e_1 + \pi_1 e_0 \wedge e_2 + \pi_2 e_0 \wedge e_3 + \pi_3 e_2 \wedge e_3 + \pi_4 e_3 \wedge e_1 + \pi_5 e_1 \wedge e_2.$$

De la même façon, nous retrouvons aussi l'opérateur *side* noté \odot (*cf.* définition 2.4). Pour cela, il faut expliciter l'isomorphisme δ de $\bigwedge^2(\mathbb{R}^4)$ vers $\bigwedge^2(\mathbb{R}^4)$, qui en pratique exprime

la dualité entre cet espace et lui-même⁷. Nous définissons δ ainsi :

$$\begin{aligned}\delta(e_0 \wedge e_1) &= e_2 \wedge e_3, & \delta(e_2 \wedge e_3) &= e_0 \wedge e_1, \\ \delta(e_0 \wedge e_2) &= e_3 \wedge e_1, & \delta(e_3 \wedge e_1) &= e_0 \wedge e_2, \\ \delta(e_0 \wedge e_3) &= e_1 \wedge e_2, & \delta(e_1 \wedge e_2) &= e_0 \wedge e_3.\end{aligned}$$

Soient les droites Π et Π' ; leur orientation est :

$$\begin{aligned}\Pi \wedge \Pi' &= (\pi_0 e_0 \wedge e_1 + \pi_1 e_0 \wedge e_2 + \pi_2 e_0 \wedge e_3 + \pi_3 e_2 \wedge e_3 + \pi_4 e_3 \wedge e_1 + \pi_5 e_1 \wedge e_2) \\ &\wedge (\pi'_0 e_0 \wedge e_1 + \pi'_1 e_0 \wedge e_2 + \pi'_2 e_0 \wedge e_3 + \pi'_3 e_2 \wedge e_3 + \pi'_4 e_3 \wedge e_1 + \pi'_5 e_1 \wedge e_2) \\ &= (\pi_0 \pi'_3 + \pi_1 \pi'_4 + \pi_2 \pi'_5 + \pi_3 \pi'_0 + \pi_4 \pi'_1 + \pi_5 \pi'_2) e_0 \wedge e_1 \wedge e_2 \wedge e_3 \\ &\sim \Pi \cdot \delta^{-1}(\Pi') = \Pi \odot \Pi',\end{aligned}$$

en vertu de l'anticommutativité du produit extérieur. Donc, l'orientation entre les deux droites correspond bien à l'opérateur *side*. De plus, l'orientation dans l'algèbre de Grassmann correspond à un produit scalaire (ici $\Pi \cdot \delta^{-1}(\Pi')$ en considérant Π' comme une $(n-2)$ -face).

Cela nous permet de conclure que l'opérateur *side* est, dans l'espace de Plücker, défini pour toute droite, réelle ou imaginaire.

3.3.1.5 Discussion

Bien sûr, la formalisation d'un espace de droites en dimension $n \geq 2$ nous permet de retrouver l'espace de Plücker. Ce résultat attendu peut sembler banal. Son intérêt réside dans le fait que, en raison de l'aspect algébrique de l'algèbre de Grassmann, il est possible de réaliser simplement des démonstrations, et donc d'offrir une meilleure connaissance d'un espace de droites en dimension n en général, et de l'espace de Plücker en particulier. Notamment, le premier résultat est que la classification des droites correspond à un produit scalaire, en dimension n , et donc fonctionne avec des droites réelles ou imaginaires. Un tel résultat n'est pas envisageable en utilisant les coordonnées de Plücker classiques ; la structure d'algèbre, et donc la base de $\bigwedge^2(\mathbb{R}^{n+1})$, jouent un rôle essentiel dans nos preuves.

En réalité, l'apport va au delà de ce premier résultat. Notamment, nous savons maintenant que bien que l'espace des droites soit projectif, une droite de \mathfrak{G}_n est en réalité une demi-droite vectorielle (pour conserver son orientation) dans le sous-espace vectoriel des 2-vecteurs. Ainsi, nous n'utilisons plus la normalisation d'une droite par l'une de ses coordonnées, telle qu'elle est effectuée par Nirenstein ; en effet, non seulement cela engendre des instabilités numériques, mais en plus cette manipulation paraît aberrante avec les droites orientées.

Enfin, cette formalisation d'un espace de droites nous a surtout permis d'unifier les méthodes de calcul de visibilité indépendamment de la dimension de l'espace géométrique. Ainsi, les droites traversant m $(n-1)$ -faces convexes de l'espace géométrique \mathfrak{G}_n sont-elles toujours, pour $n \geq 2$, définies comme celles ayant une orientation consistante (soit toutes positives, soit toutes négatives) [Aveneau 2011].

7. Avec $n = 3$ le domaine et le codomaine sont les mêmes, ce qui est faux pour $n = 2$ et $n \geq 4$.

Le paragraphe suivant s'intéresse au problème du polytope minimal, essentiel pour la présentation de nos nouveaux algorithmes pour la visibilité.

3.3.2 Le polytope minimal

Le polytope minimal est un problème majeur dans les méthodes de Nirenstein ou de Bittner. En pratique, l'absence de sa définition correcte empêche un calcul qui soit exact à 100%. Nous en donnons ici une caractérisation fonctionnant en toute dimension $n \geq 2$.

3.3.2.1 La problématique

Notre théorème 3.1 décrit l'ensemble des droites traversant deux (et par extension m) $(n - 1)$ -faces convexes de \mathfrak{G}_3 en configuration non dégénérée. Il pose cependant deux questions : d'abord, il ne fonctionne que dans \mathfrak{G}_3 , et ne permet pas une généralisation d'une méthode de calcul de visibilité en dimension quelconque ; ensuite, il ne décrit pas l'ensemble des droites traversant deux faces convexes dans les cas dégénérés.

Rappelons qu'une dégénérescence apparaît lorsque le plan support d'une des $(n - 1)$ -faces coupe l'autre. Dans ce cas, les travaux de Nirenstein [Nirenstein 2003], de Bittner [Bittner 2002] ou d'Haumont [Haumont 2005] nécessitent un *rognage* de la face découpée, permettant ensuite d'appliquer les méthodes sur deux voire quatre couples de faces (cf. §2.4.2.2). Cette solution introduit des erreurs dans le calcul de la visibilité : en effet, celle-ci est calculée sur des morceaux de faces rognées, *i.e.* un ensemble de points géométriques et donc de droites est perdu dans l'opération. Si la visibilité n'a lieu qu'en tenant compte de certains de ces points, alors un calcul donne à tort un résultat négatif à la simple question "deux faces se voient-elles ?". Si dans les applications de Nirenstein, Bittner ou Haumont, cette approximation ne pose pas réellement de problèmes, pour les nôtres elle est au contraire inadmissible.

Heureusement, l'expression d'un espace de droites via l'algèbre de Grassmann nous permet de caractériser le problème, et surtout de le résoudre.

3.3.2.2 Théorème du polytope minimal

Nous commençons par proposer une définition formelle du polytope minimal. Soient \mathbf{A} et \mathbf{B} deux $(n - 1)$ -faces convexes de \mathfrak{G}_n pour $n \geq 2$, de sommets respectifs a_1, \dots, a_q et b_1, \dots, b_r .

Définition 3.11 *Le polytope minimal, noté $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$, et représentant dans \mathfrak{D}_n l'ensemble $\mathcal{L}_{\mathbf{A}}^{\mathbf{B}}$ des droites traversant \mathbf{A} puis \mathbf{B} , est le polytope convexe ayant les propriétés suivantes :*

1. $\mathcal{L}_{\mathbf{A}}^{\mathbf{B}} \subseteq \mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$.
2. $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}} \cap G^{\mathbb{R}}(2, n + 1) \subseteq \mathcal{L}_{\mathbf{A}}^{\mathbf{B}}$.
3. Si $\mathcal{P}_{\mathbf{A}}^{\mathbf{B}}$ est un ensemble convexe dans \mathfrak{D}_n tel que $\mathcal{L}_{\mathbf{A}}^{\mathbf{B}} \subseteq \mathcal{P}_{\mathbf{A}}^{\mathbf{B}}$, alors $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}} \subseteq \mathcal{P}_{\mathbf{A}}^{\mathbf{B}}$.

Les deux premières propriétés signifient que le polytope minimal \mathcal{M}_A^B est un représentant de \mathcal{L}_A^B dans l'espace des droites \mathcal{D}_n ; ainsi, une droite l traverse **A** puis **B** si et seulement si sa représentation dans \mathcal{D}_n est contenue dans \mathcal{M}_A^B . La troisième propriété indique que \mathcal{M}_A^B est le polytope minimal : il n'existe pas de plus petit polytope convexe représentant \mathcal{L}_A^B et contenant \mathcal{M}_A^B .

Le théorème suivant donne une caractérisation du polytope minimal pour deux faces dans une configuration canonique précise, et stipule qu'il n'existe aucun polytope minimal dans les autres configurations.

Théorème 3.4 *Soit \mathcal{H}_A et \mathcal{H}_B les deux plans supports des $(n-1)$ -faces **A** et **B** respectivement. Si \mathcal{H}_A (resp. \mathcal{H}_B) ne coupe pas la face **B** (resp. **A**), ou bien seulement sur son bord, alors le polytope minimal \mathcal{M}_A^B est l'enveloppe convexe des droites $l_{ij} = a_i \wedge b_j$, $(i, j) \in [1 \dots q] \times [1 \dots r]$ reliant les sommets de **A** à ceux de **B**.*

*Sinon, l'ensemble des droites traversant **A** puis **B** ne peut pas être représenté par un unique polytope convexe dans \mathcal{D}_n .*

3.3.2.3 Discussion

La preuve de théorème 3.4 est donnée dans [Aveneau 2011], et repose sur l'algèbre de Grassmann et les différentes définitions et propriétés que nous avons pu montrer au préalable.

Il est intéressant de noter que ce théorème généralise le théorème 3.1, qui ne fonctionne que pour des droites traversant deux polygones convexes de \mathcal{G}_3 : la nouvelle version porte sur les droites traversant deux polytopes convexes de \mathcal{G}_n , pour $n \geq 2$.

Le début du théorème 3.4 reprend et généralise un point fort du théorème 3.1, en supprimant toute notion de rognage des faces dans les cas dégénérés. Les faces sont seulement coupées suivant un hyperplan, permettant un calcul exact.

La deuxième partie du théorème 3.4 est fondamentalement nouvelle : elle dit sans ambiguïté que lorsque l'hyperplan \mathcal{H} support d'une des deux faces coupe l'autre face **F** en *son intérieur*, alors il n'y a pas de polytope minimal ; autrement dit, l'orientation des droites n'est plus consistante, et il devient obligatoire de découper **F** par \mathcal{H} .

En résumé, en découpant toutes les faces par tous les hyperplans supports, alors un problème donné est transformé sans perte d'information en un ensemble de problèmes pour lesquels il existe toujours un polytope minimal. Le calcul de visibilité ou d'occultation devient alors théoriquement exact en toute configuration.

3.4 Apports algorithmiques

Nous discutons ici des différentes méthodes pratiques permettant de calculer la visibilité dans \mathcal{G}_3 . Nous commençons par les travaux sur le calcul d'occultation, qui utilisent directement les résultats obtenus sur l'espace de droites. Puis, nous proposons une méthode originale de calcul de visibilité à la demande.

3.4.1 Calcul d'occultation dans \mathcal{D}_3

Nous décrivons ici l'application directe des résultats théoriques obtenus dans la section 3.3, résumant une partie des travaux de thèse de Sylvain Charneau [Charneau 2007a, Charneau 2007b, Aveneau 2011].

3.4.1.1 Initialisation d'un calcul

Partant d'un espace de droites défini algébriquement, nous proposons une méthode permettant de calculer l'occultation en dimension n , appliquée au cas $n = 3$. Comme pour nos premiers travaux sur l'occultation dans \mathcal{G}_3 (cf. §3.1.2) basés sur ceux de Bittner [Bittner 2002], notre méthode repose sur un arbre BSP. Cet arbre encode les droites traversant les $(n - 1)$ -faces \mathbf{A} et \mathbf{B} , *i.e.* des polygones convexes en dimension 3.

Notre méthode utilise le théorème 3.4 en de nombreuses circonstances. Puisque nous voulons traiter la visibilité sans erreur, il n'est pas question d'utiliser le rognage de Nirenstein (cf. §2.4.2.2). Lorsque le plan support de \mathbf{A} coupe \mathbf{B} en son intérieur, et/ou inversement, une première idée est naturellement de construire plusieurs arbres. Cependant, en ajoutant dès la racine d'un unique arbre la droite correspondant à l'intersection des plans supports de \mathbf{A} et \mathbf{B} , alors nous obtenons deux polytopes encodant la visibilité d'un côté donné de \mathbf{A} .

Pour la construction de l'arbre, notre première méthode utilise la représentation complète des polytopes contenant les droites traversant \mathbf{A} puis \mathbf{B} mais pas certains occulteurs. Cela inclut la topologie liant des faces de dimensions 0 à 4 dans l'espace des droites \mathcal{D}_3 (projectif, et de dimension 6), ainsi que leur géométrie. Si nous utilisons cette représentation dans le paragraphe 3.1.2, c'est pour effectuer correctement un rejet complet des occulteurs par les hyperplans des polytopes, et inversement des polytopes par les hyperplans des occulteurs ; clairement, il faut à la fois les 0 et les 4-faces. Néanmoins, Haumont *et al.* [Haumont 2005] ont proposé d'utiliser une version simplifiée du polytope, ne conservant que les 0 et 1 faces, et leurs incidences aux 4-faces. Nous réutilisons cette technique de *squelettes*, car évitant la mise à jour complète du graphe d'un polytope après une découpe elle est plus rapide, mais aussi parce qu'elle élimine un certain nombre d'erreurs numériques.

Malgré tout, et afin d'obtenir un calcul sûr, nous utilisons la bibliothèque de calculs en précision étendue GMP [GNU 2006]. Elle permet d'augmenter la précision numérique, repoussant les problèmes numériques qui néanmoins persistent. Aussi, nous utilisons une normalisation des coordonnées des droites : une droite étant un vecteur, cela ne change pas sa nature si elle est multipliée par un réel positif. Cette normalisation est effectuée sur une hypersphère dans l'espace des 2-vecteurs, et évite de trop grandes différences entre les valeurs à comparer lors de la classification.

3.4.1.2 Rejet d'un polytope

L'insertion d'un occulteur dans l'arbre de visibilité repose sur le calcul du polytope minimal associé. Bien entendu, nous utilisons le polytope résultant de l'enveloppe convexe des

droites reliant tous les sommets d'une première face à tous les sommets d'une seconde. Notons que le squelette et le graphe d'incidence associés ont une topologie décidée uniquement par le nombre de sommets des deux faces, et donc facile à initialiser. En pratique, afin de réduire au maximum la géométrie d'un polytope minimal, nous découpons le polygone **A** et l'occulteur de sorte à ne conserver que des droites traversant réellement d'abord **A**, puis l'occulteur et enfin **B**.

Contredisant l'intuition première, utiliser l'orientation (ou un autre critère) pour choisir le prochain occulteur n'améliore pas en général les performances, et parfois même les dégrade. En effet, la complexité visuelle n'est pas déterminée par le nombre d'occulteurs, mais par leurs impacts en terme d'occultation : ainsi le choix d'un certain occulteur peut parfois éliminer une large proportion des droites entre **A** et **B** ; sa suppression (parce qu'il est mal orienté, qu'il est trop loin ou trop proche, qu'il est trop petit ...) est alors une mauvaise stratégie.

Le rejet d'un polytope, qui permet d'éviter de descendre un occulteur dans un sous-arbre inutilement et donc de sur-découper les données, est une version encore améliorée du rejet que nous avons proposé initialement dans le paragraphe 3.1.2. Reprenant l'idée d'un double test de rejet, nous l'étendons de façon à rejeter chaque arête d'un occulteur. Ainsi les polytopes qui descendent dans un arbre donné auront deux sortes d'arêtes : celles qui ne sont pas rejetées, et donc candidates pour devenir un noeud interne de l'arbre de visibilité ; celles qui ont été rejetées, et qui ne sont pas utilisées pour découper la visibilité. Cette méthode permet donc de rejeter des polytopes complètement ou partiellement, réduisant encore la taille des arbres obtenus.

3.4.1.3 Résultats et Discussion

Les résultats sont présentés dans [Charneau 2007b, Charneau 2007a]. Les scènes utilisées sont illustrées sur la figure 3.2. Elles sont de complexité géométrique relativement faible (quelques centaines de polygones à 19 milliers). Néanmoins, ce qui compte réellement est la complexité visuelle, qui n'est que rarement en relation directe avec la première.

Les résultats exhibent clairement que notre nouvelle méthode de calcul d'occultation reposant sur l'espace de droites \mathcal{D}_3 réduit de beaucoup la taille des arbres de visibilité obtenus.

Par exemple, la scène "Un crayon" est composée de 339 polygones principalement localisés dans la mine. Elle a ainsi localement une forte complexité visuelle. Un rejet partiel de Nirenstein engendre un arbre composé de 47851 noeuds, et cela en 1 heure 36 minutes sur un machine de test d'époque ; notre rejet total réduit l'arbre à 1811 noeuds en 3 minutes 25 de calcul. Ici, l'intérêt ne réside pas dans les temps de calcul qui restent raisonnables dans les deux cas, s'agissant d'un traitement amont (les arbres sont sauvegardés dans des fichiers). L'intérêt est dans la réduction de la taille de l'arbre : en prenant notre nouvelle méthode comme référence (1.00), alors nos travaux précédents ont une taille de 1.27 (2308 noeuds), et un rejet de type Nirenstein 26.422.

Même sans utiliser la silhouette, le nouveau rejet permet de réduire la taille des arbres par

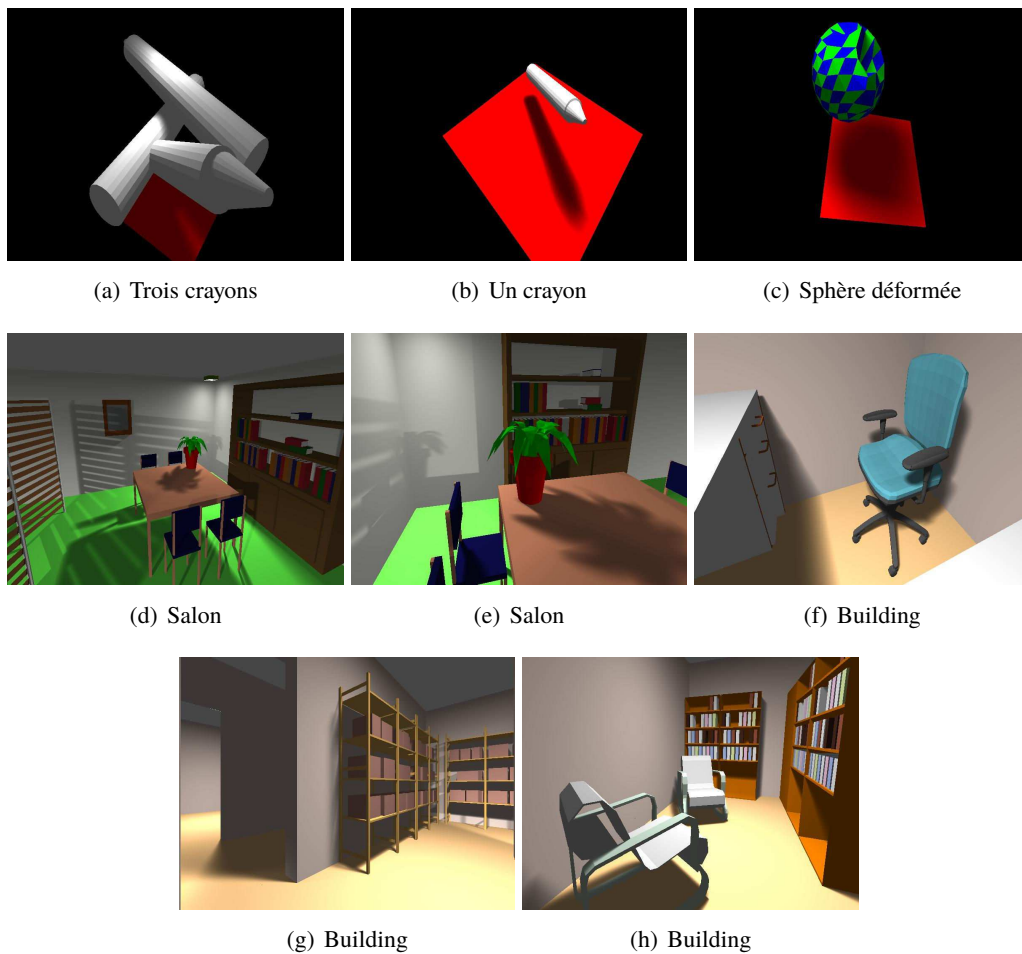


FIGURE 3.2 – Scènes de tests utilisées dans [Charneau 2007a].

rapport à nos précédents travaux décrit dans le paragraphe 3.1.2. Ainsi sur la scène “Sphère déformée” contenant 323 polygones, la taille des arbres en versions avec ou sans silhouette, ancienne version ou version de Nirenstein sont respectivement de 1701, 4160, 11030 et 49697 noeuds. Clairement, l’utilisation de la silhouette est efficace ; mais cela illustre aussi que le nouveau test de rejet est avantageux par rapport à notre première version (facteur de 2.65).

Notre méthode est donc meilleure en terme de qualité à plus d’un titre :

- elle supprime les erreurs dues au rognage et donc est exacte,
- les arbres de visibilité résultants sont plus petits.

Néanmoins il existe encore un problème, en lien avec le nombre d’occludeurs utilisables. La scène “Building” est constituée de 18865 polygones. Lors du calcul de tous les arbres de visibilité, le nombre maximum d’occludeurs est de 10986. Là où la méthode de Nirenstein n’arrive plus à calculer aucun arbre, la nôtre produit un résultat de qualité, permettant une exploitation pour le rendu avec ombre douce (*cf.* chapitre 4). Néanmoins, le nombre de découps dans un arbre commence à impacter la qualité des calculs numériques. Il est difficile

d'utiliser des scènes plus complexes en pré-calcul.

Un solution pour réduire cette complexité s'inspire directement d'une des utilisations que nous avons faites d'un arbre de visibilité : l'idée est de calculer la visibilité d'une boîte englobant un objet quelconque ; cette boîte est ensuite déplacée dans une scène quelconque, pour avoir un rendu dynamique avec ombres douces. En utilisant des calculs hiérarchiques de visibilité sur des portions de scènes, alors il est tout à fait possible de s'affranchir de la complexité géométrique (et donc visuelle), et donc de calculer puis d'exploiter des arbres de visibilité sur des scènes de tailles importantes.

3.4.2 Calcul à la demande

Le calcul d'un arbre de visibilité complet est un processus relativement lourd, qui empêche notamment la modification de la géométrie. Dans le but de rendre ce calcul plus interactif, nous nous sommes aperçus que l'information calculée est en pratique rarement utilisée. Cette section décrit une seconde méthode de calcul de visibilité s'inspirant de l'évaluation paresseuse. Elle a été conçue entre les années 2008 et 2011, en collaboration avec Frédéric Mora, maître de conférences à l'Université de Limoges, membre du département DMI de l'institut XLIM ; elle a été publiée dans [Mora 2012, Apostu 2012, Mora 2013].

3.4.2.1 Motivations

Le calcul d'un arbre de visibilité par couple de faces reste un processus particulièrement coûteux à la fois en temps et en mémoire. Par exemple, lors d'un rendu avec ombres douces il est nécessaire de calculer par source lumineuse autant d'arbres qu'il y a d'autres polygones dans la scène. Avec une scène contenant un million de polygones et une unique source lumineuse, il faut alors un million d'arbres de visibilité. Chaque arbre occupant en mémoire potentiellement quelques Mo, il devient évident que l'utilisation de toutes ces informations nécessite une bonne gestion de la mémoire.

Dès lors, il est important de justifier une telle masse de calculs et de données en fonction de l'application visée. Or, nos applications de tests sont généralement pour du rendu, comme dans le chapitre 4. Dans un tel cas, l'exploitation de l'information de visibilité est effectuée depuis un point de vue donné, par exemple sur une face éclairée dans un calcul d'illumination locale (cf. section 4.1). Il nous est vite apparu que l'utilisation pratique des données dans ce genre de cas est particulièrement restreinte.

Cette constatation nous amène à considérer un calcul uniquement lorsque c'est nécessaire, suivant une approche paresseuse (ou retardée, ou encore un calcul à la demande). Une telle méthode est généralement utilisée pour manipuler des données infinies, comme les éléments d'une suite quelconque. Plutôt que de tenter (sans succès dans ce cas) de pré-calculer la suite, l'évaluation paresseuse retarde au maximum les calculs effectifs au moment de leur première utilisation.

Dans notre cas, l'idée est donc de développer l'arbre de visibilité lorsque, et uniquement dans ce cas, il est utilisé effectivement pour la première fois. L'avantage de la méthode est

qu'elle est numériquement beaucoup plus stable : il n'y a plus de polytope explicite, mais uniquement implicite. Donc nous n'utilisons plus de bibliothèque de calcul telle que GMP, et les calculs sont alors beaucoup plus rapides.

Pour mener à bien ce nouvel algorithme, nous sommes repartis des travaux précédents sur le partitionnement de l'espace des droites \mathcal{D}_n [Pellegrini 1991]. Nous considérons donc une partition de \mathcal{D}_n , effectuée en utilisant les bords (les arêtes) des occulteurs entre une source \mathbf{A} et une cible \mathbf{B} . Dans une telle partition, chaque cellule représente un ensemble de droites traversant exactement les mêmes convexes de l'espace géométrique \mathcal{G}_n , *i.e.* \mathbf{A} puis une suite d'occulteurs (\mathbf{O}_i) puis \mathbf{B} . Toute droite est dans une unique cellule de la partition de l'espace : elle traverse une unique liste de polytopes dans l'espace géométrique.

Bien sûr, travailler dans une partition de l'espace de droites peut sembler des plus contradictoires : c'est exactement ce que tous les travaux précédents ont cherché à éviter depuis Pu [Pu 1999]. Pourtant, cette méthode se révèle extrêmement efficace en pratique, puisque la plupart des cellules d'une partition de l'espace des droites ne sert strictement à rien : une faible proportion seulement contient des droites réelles.

3.4.2.2 Présentation du calcul à la demande

Soit deux faces convexes \mathbf{A} et \mathbf{B} , et soient les occulteurs (\mathbf{O}_i) contenus dans l'enveloppe convexe de \mathbf{A} et \mathbf{B} . Pour calculer la visibilité entre \mathbf{A} et \mathbf{B} , nous nous intéressons aux droites réelles entre \mathbf{A} et \mathbf{B} qui, soit ne traversent aucun occulteur \mathbf{O}_i , soit en traversent au moins un. Soit une partition de l'espace des droites \mathcal{D}_n construite en utilisant les $(n - 2)$ -faces bordant les occulteurs. Les cellules correspondent à des *classes d'équivalence* de droites, qui traversent toutes les mêmes occulteurs. Nous définissons alors :

- Une *classe visible* : toute classe d'équivalence qui représente un ensemble de droites ne traversant aucun occulteur.
- Une *classe invisible* : toute classe d'équivalence correspondant à un ensemble de droites traversant au moins un occulteur.
- Une *classe indéfinie* : une classe d'équivalence pas encore classée *visible* ou *invisible*.

L'arbre de visibilité construit à la demande fournit une représentation des classes d'équivalence induites par les différents occulteurs (\mathbf{O}_i) . Chaque feuille représente l'une de ces trois classes. La construction est effectuée à la demande, lorsqu'une requête de visibilité arrive sur une feuille indéfinie. Elle se base sur deux opérations : l'insertion d'un occulteur, et la croissance de l'arbre. La première détermine les feuilles de l'arbre impactées par un occulteur donné. La seconde consiste à remplacer une feuille indéfinie par l'arbre déduit d'un occulteur donné ; en pratique un tel arbre BSP est relativement simple à calculer (*cf.* figure 3.3).

Notre structure de données est relativement simple : un noeud est étiqueté par la nature de la classe qu'il représente. Un noeud indéfini contient une liste d'occulteurs (\mathbf{O}_i) non encore traités (ici la taille de la liste dépend du noeud). L'insertion consiste donc à choisir un occulteur \mathbf{O}_k , à remplacer le noeud courant par l'arbre représentant l'occulteur choisi (*cf.* figure 3.3), puis à faire descendre les autres occulteurs (\mathbf{O}_i) pour $i \neq k$ dans le nouveau

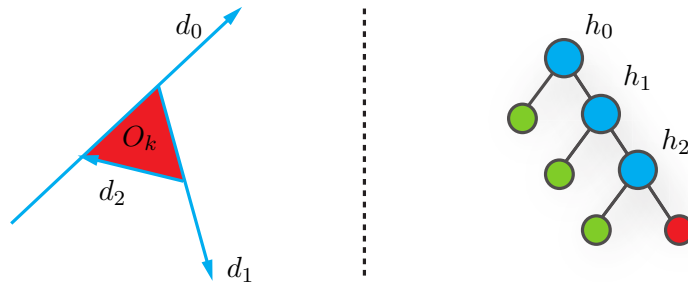


FIGURE 3.3 – Soient d_0, d_1, d_2 les droites supports des arêtes d'un occulteur \mathbf{O}_k , et soient h_0, h_1 et h_2 leurs duaux dans l'espace de droites \mathfrak{D}_n . L'arbre représente les quatre classes d'équivalence générées par les arêtes de \mathbf{O}_k : trois sont classées visibles (feuilles de gauche), la dernière invisible (feuille de droite). Une telle représentation est notée $bsp(\mathbf{O}_k)$.

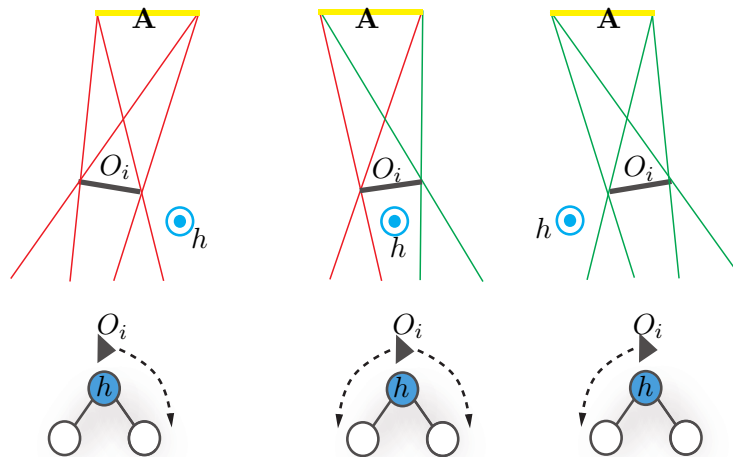


FIGURE 3.4 – Insertion d'un occulteur \mathbf{O}_i dans \mathfrak{G}_2 : \mathbf{O}_i traverse un sous-ensemble de toutes les droites sortant de \mathbf{A} . Cet ensemble peut avoir une orientation négative (à gauche), positive (à droite) ou mixte (au centre) par rapport à un hyperplan h donné. Le théorème 3.4 permet de définir cette orientation en utilisant l'enveloppe convexe des droites reliant un sommet de \mathbf{A} à un sommet de \mathbf{O}_i . Ce rejet permet de descendre \mathbf{O}_i à gauche, à droite ou dans les deux sous-arbres d'un noeud.

sous-arbre (cf. figure 3.4). La descente des occulteurs dans l'arbre BSP représentant \mathbf{O}_k repose sur le théorème 3.4.

Le choix d'un occulteur idéal est un problème très complexe. Nos tests nous ont convaincu qu'une décision purement aléatoire est la meilleure solution pratique.

3.4.2.3 Résultats et discussion

Notre méthode, bien que paresseuse, se révèle particulièrement rapide, comme nous le verrons dans le chapitre 4, §4.1.3. Notamment, elle tient ses promesses, à savoir qu'elle évite l'exploration de portions d'un arbre de visibilité qui ne sont pas utilisées. Il existe deux raisons justifiant cette non utilisation : ne pas en avoir besoin, et le fait qu'une telle région ne contient aucune droite réelle. Cette dernière est une des raisons principales ayant motivé cette version paresseuse.

Par contre, cette méthode souffre de problèmes similaires aux approches précédentes, liés à la complexité théorique maximale d'un arrangement d'hyperplans dans l'espace de droites ($\mathcal{O}(n^4 \log n)$ suivant [Pellegrini 1993]). Certaines configurations visuelles peuvent ainsi nécessiter des arbres de très grandes tailles, dépassant parfois les limites physiques des machines.

La figure 3.5 montre le comportement typique d'une requête dans \mathcal{G}_3 , en terme d'évolution de la taille de l'arbre et du coût de calcul. Il s'agit d'une requête extraite de [Mora 2012], et travaillant sur une scène bien connue en synthèse d'images (le modèle "conférence", comportant 280 milliers de polygones). Il y a 2896 occulseurs dans l'enveloppe convexe entre la source A et la cible B, et près de 18000 requêtes sont effectuées. Beaucoup d'entre elles nécessitent un développement d'une nouvelle partie de l'arbre.

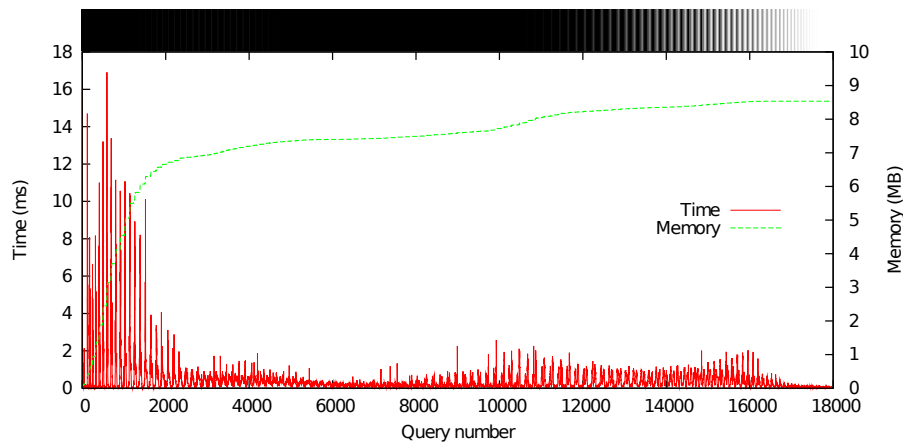


FIGURE 3.5 – Construction paresseuse d'un arbre BSP représentatif. L'abscisse correspond au nombre de requêtes. L'ordonnée gauche est le temps en millisecondes, et correspond à la courbe continue (rouge). L'ordonnée droite indique la consommation mémoire en Mo, et correspond à la courbe croissante hachurée (verte). Au dessus du graphique, une illustration en niveau de gris de la quantité de visibilité retournée par la requête : du noir au blanc indique de 0 à 100% visible.

Au début, l'arbre croît rapidement, car les requêtes requièrent de nouvelles données non encore calculées. Ensuite, l'arbre grossit doucement, uniquement par petits pas : les données déjà calculées sont largement réutilisées. C'est le comportement typique d'une requête de

visibilité en général. Nous avons donc des requêtes coûteuses, et d'autres économiques. L'efficacité de notre algorithme repose sur un bon ratio des dernières sur les premières.

Bien entendu, cette évaluation paresseuse nécessite généralement un ordonnancement intelligent des requêtes afin d'assurer une certaine cohérence dans les accès à la mémoire. L'idée sous-jacente est de réutiliser les portions d'arbres déjà chargées dans les différents niveaux de cache d'un processeur moderne (*e.g.* niveaux L1, L2 et L3 sur un Intel Core i7).

3.5 Conclusion

Ce chapitre décrit nos différentes contributions concernant la visibilité en dimensions 2, 3 et au delà. Nos premières études, dans le cadre des thèses de Pierre Combeau et Frédéric Mora, nous ont convaincu de la nécessité d'un travail plus approfondi sur la notion d'espace de droites. La thèse de Sylvain Charneau a permis des avancées notables sur la compréhension d'un espace de droites, en dimension quelconque.

Cette compréhension indépendamment de la dimension de l'espace géométrique est particulièrement importante pour permettre la réutilisation d'une méthode dans un espace de dimension différente, et ce à moindre coût. En effet, l'histoire nous a montré que l'application des méthodes de \mathfrak{G}_2 à \mathfrak{G}_3 a été bien difficile, le changement d'espace de droites ayant conduit à bien des erreurs d'interprétation et de raisonnement.

Une définition algébrique de \mathfrak{D}_n permet des raisonnements formels. Nous avons ainsi prouvé plusieurs propositions, notamment le théorème 3.4 définissant exactement l'ensemble des droites traversant deux faces convexes de \mathfrak{G}_n . Ce résultat a déjà eu de nombreuses utilisations, dans deux nouvelles méthodes de calcul de visibilité.

La première méthode calcule exactement l'occultation entre deux convexes **A** et **B**. Contrairement aux travaux précédents, elle est exacte, car elle gère correctement les différents problèmes de dégénérescences et autres cas particuliers. La seconde méthode en est une version paresseuse : elle permet d'obtenir un résultat à la demande, et évite ainsi de surcharger la mémoire après de longs calculs inutiles.

Ces deux méthodes ont été appliquées en dimension 3. Notre objectif est de les utiliser dans des espaces de dimensions supérieures, par exemple un espace de dimension 4, englobant \mathfrak{G}_3 et le temps. Calculer la visibilité dans un tel espace nous permettrait ainsi de définir des ensembles de droites (des classes d'équivalence) stables durant un instant donné, afin de réaliser des simulations de qualité en environnement dynamique.

Un autre piste de travaux concerne la complexité de la visibilité. Dans certains cas, la complexité de nos méthodes tend vers celle d'un arrangement d'hyperplans dans l'espace de droites. Le résultat est alors une explosion des temps de calculs et de la mémoire nécessaire (même avec une méthode paresseuse). Pour éviter cela, nous travaillons actuellement sur une nouvelle représentation de l'information de visibilité. L'une de ses caractéristiques principales est que la visibilité peut être calculée en place, *i.e.* dans un tableau de m cases pour m occulteurs, laissant entrevoir des applications sur GPU.

Travaux en Informatique Graphique

Sommaire

4.1 Travaux sur les ombres douces	63
4.1.1 Introduction au calcul d'ombre douce	63
4.1.2 Méthode à base de pré-calcul d'occultation	66
4.1.3 Méthode à base d'évaluation paresseuse	70
4.2 Autres applications de l'évaluation paresseuse	72
4.2.1 Réduction des biais dans les cartes de photons	72
4.2.2 Occlusion ambiante	78
4.3 Autres travaux en informatique graphique	83
4.3.1 Lobes de cosinus	83
4.3.2 Visualisation interactive de bâtiments	88
4.4 Conclusion	92

Ce chapitre présente nos différents travaux dans le domaine du rendu en synthèse d'images. Pour une large partie, ces travaux utilisent nos contributions sur la visibilité, présentées dans le chapitre 3. Le calcul d'occultation est ainsi appliqué aux ombres douces, à la réduction des biais dans les cartes de photons, et au calcul de l'occlusion ambiante. La dernière partie de ce chapitre est consacrée à deux autres contributions, les lobes de cosinus et le rendu dans des environnements architecturaux.

4.1 Travaux sur les ombres douces

4.1.1 Introduction au calcul d'ombre douce

4.1.1.1 La problématique

Beaucoup de travaux antérieurs traitent du calcul d'une image avec ombres douces : à partir de subdivision de surfaces [Atherton 1978], avec le volume d'ombres [Crow 1977, Bergeron 1986, Chin 1989, Heidmann 1991, Everitt 2003, Lloyd 2004], les méthodes stochastiques [Cook 1984, Cook 1986], l'utilisation de faisceaux [Amanatides 1984, Heckbert 1984, Ghazanfarpour 1998], les volumes d'ombres et de pénombres [Nishita 1985a, Nishita 1985b, Chin 1992, Tanaka 1997, Laine 2005a, Forest 2006, Forest 2008, Forest 2009], ou encore les silhouettes étendues et le volume d'ombre douce [Akenine-Möller 2002, Laine 2005b].

Le calcul d'une image de synthèse consiste à résoudre avec plus ou moins de rigueur l'équation de transfert lumineux (ou ETL, cf. Annexe A.3, équation (A.9) page 143) :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_A L_i(x, \mathbf{y} \rightarrow \mathbf{x}) f(x, \mathbf{y} \rightarrow \mathbf{x}, \mathbf{o}) G(x, y) V(x, y) dA_y,$$

avec :

– le terme géométrique $G(x, y) = \frac{|\mathbf{n}_x \cdot \mathbf{x} \rightarrow \mathbf{y}| |\mathbf{n}_y \cdot \mathbf{y} \rightarrow \mathbf{x}|}{\|x - y\|^2}$,

– l'ensemble A des surfaces de la scène,

– le terme de visibilité $V(x, y)$ indiquant si x et y sont mutuellement visibles.

Cette équation indique que la luminance émise en un point x donné, dans une direction \mathbf{o} donnée, correspond à :

– une partie auto émise, prenant donc en compte les sources lumineuses,

– et une partie réfléchie, correspondant aux produits de la FDRB par la luminance incidente pour toutes les directions de l'hémisphère entourant x .

Cette équation est récursive, puisque L_o , la luminance émise égale L_i , celle reçue (cf. §A.1.3.1). Il vient (cf. § A.3.3) :

$$\begin{aligned} L_o(x, \mathbf{o}) &= L_e(x, \mathbf{o}) + \int_A L_o(y, \mathbf{y} \rightarrow \mathbf{x}) f(x, \mathbf{y} \rightarrow \mathbf{x}, \mathbf{o}) G(x, y) V(x, y) dA_y \\ &= L_e(x, \mathbf{o}) + \int_A L_e(y, \mathbf{y} \rightarrow \mathbf{x}) f(x, \mathbf{y} \rightarrow \mathbf{x}, \mathbf{o}) G(x, y) V(x, y) dA_y \\ &\quad + \int_A \int_A L_i(y, \mathbf{z} \rightarrow \mathbf{y}) f(y, \mathbf{z} \rightarrow \mathbf{y}, \mathbf{y} \rightarrow \mathbf{x}) G(y, z) V(y, z) dA_z \\ &\quad \quad \quad f(x, \mathbf{y} \rightarrow \mathbf{x}, \mathbf{o}) G(x, y) V(x, y) dA_y \\ &= L_{\text{émis}} + L_{\text{direct}} + L_{\text{indirect}} \end{aligned}$$

Le domaine de L_{direct} peut être réduit à l'ensemble A_D des surfaces ayant une émittance non nulle (i.e. $L_e \neq 0$).

$$L_{\text{direct}} = \int_{A_D} L_e(y, \mathbf{y} \rightarrow \mathbf{x}) f(x, \mathbf{y} \rightarrow \mathbf{x}, \mathbf{o}) G(x, y) V(x, y) dA_y \quad (4.1)$$

Calculer une ombre douce, c'est calculer l'éclairement direct L_{direct} dû à des sources quelconques.

4.1.1.2 Simplifications du problème

Nous distinguons plusieurs parties dans l'intégrale L_{direct} :

– l'émittance L_e dépend de la direction d'émission et de la position sur une source ;

– la FDRB dépend de la direction d'incidence, et donc est généralement variable ;

– le terme géométrique dépend complètement de la géométrie (orientation et position) ;

– le terme d'occultation $V(x, y)$ dépend de x et de tous les points y sur les sources.

Généralement cette intégrale est à son tour découpée en plusieurs parties, afin de traiter chaque source indépendamment. Dans la suite, nous simplifions le problème, pour se focaliser sur le terme d'occultation. Néanmoins les techniques proposées dans cette section sont généralisables.

Le rendu avec ombres douces considère des sources lumineuses surfaciques, comme des cylindres pour simuler des tubes néons, des sphères pour des ampoules classiques, etc. Généralement, nous utilisons des sources polygonales, par exemple issues de la triangulation d'une sphère.

Notre approche est similaire à celle de Nishita *et al.* [Nishita 1985a, Nishita 1983]. Leur méthode consiste à intégrer l'émission sur des ensembles continus et plats de points. Pour une source donnée, Nishita *et al.* considèrent que l'émittance est uniforme : la luminance émise ne dépend ni de la position y ni de la direction $y \rightarrow x$. De plus, ils ajoutent la contrainte que la réflectance est uniforme (modèle de Lambert). Ces simplifications très largement pratiquées n'ont aucun impact visuel tant que les sources utilisées sont de tailles raisonnables. La condition principale devient donc que tous les points y du domaine d'intégration soient visibles de x .

Nous utilisons une approche similaire, dans laquelle le calcul de visibilité est effectué par nos techniques issues du chapitre 3.

4.1.1.3 Méthode de référence

En un point x donné, et pour rendre une ombre douce due à l'ensemble des points sources A_D , le calcul exact de l'ensemble des points visibles depuis x est un processus coûteux. Une optimisation consiste à échantillonner A_D , et à estimer l'ombre par une méthode approchée. Avec une méthode de Monte-Carlo, l'intégrale L_{direct} est alors estimée en moyennant un ensemble de contributions ponctuelles. Pour une source lumineuse d'émittance uniforme L_e et pour une réflectance caractérisée par le terme ρ , il vient :

$$L_{\text{direct}} \approx \frac{1}{N} \sum_{i=1}^N \left\{ \frac{\rho L_e G(x, y_i) V(x, y_i)}{\text{pdf}(y_i)} \right\}$$

où (y_i) est une suite des N points aléatoirement choisis sur la surface de la source suivant la densité de probabilité $\text{pdf}(y_i)$ (cf. annexe B).

Le principal inconvénient d'une méthode de Monte-Carlo réside dans le bruit généré : l'erreur d'un tel estimateur est fonction de $1/\sqrt{N}$ (cf. §B.1.2). Cette erreur se traduit par un effet de grain dans les images (cf. figure 4.1(a)).

En augmentant le nombre d'échantillons, cet effet devient de moins en moins perceptible. Néanmoins, cela engendre aussi une augmentation des temps de calcul.

Dans la suite, la méthode de Monte-Carlo est utilisée comme référence. C'est une méthode largement répandue dans les moteurs de rendus, notamment en raison de sa robustesse et de sa souplesse (elle permet d'utiliser n'importe quel type de sources lumineuses). Notre version est optimisée de la façon suivante :

- traitement de 4 points x_i en parallèle, en utilisant les instructions vectorielles (SSE) [Wald 2004] ;
- parallélisation avec OpenMP pour utiliser les architectures multi-cœurs ;

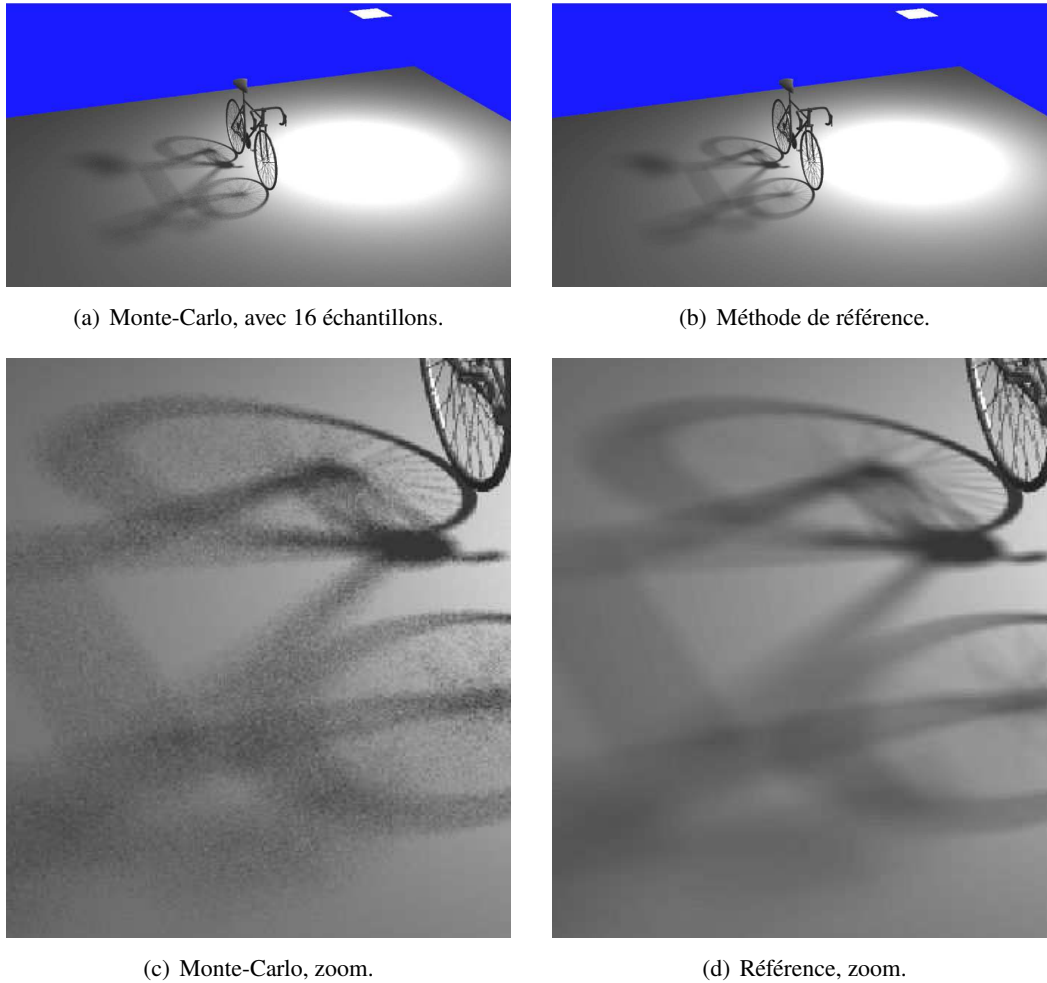


FIGURE 4.1 – Exemple de bruit dans une ombre calculée par méthode de Monte-Carlo.

- requête de visibilité point à point utilisant une structure accélératrice de type KD-tree, avec heuristique aire-surface [Havran 2001] ;

4.1.2 Méthode à base de pré-calcul d'occultation

Le rendu d'images avec ombres douces est la première méthode graphique que nous avons étudiée, comme sujet d'application du calcul d'occultation dans un espace de droites de dimension 3. Par rapport aux travaux précédents [Nirenstein 2002, Bittner 2002], le rendu d'une ombre douce exploite beaucoup plus d'informations issues du pré-calcul, et justifie pleinement l'utilisation d'un arbre de visibilité. De plus, il permet de valider une méthode de calcul d'occultation, puisque toute erreur est très rapidement perceptible par l'oeil.

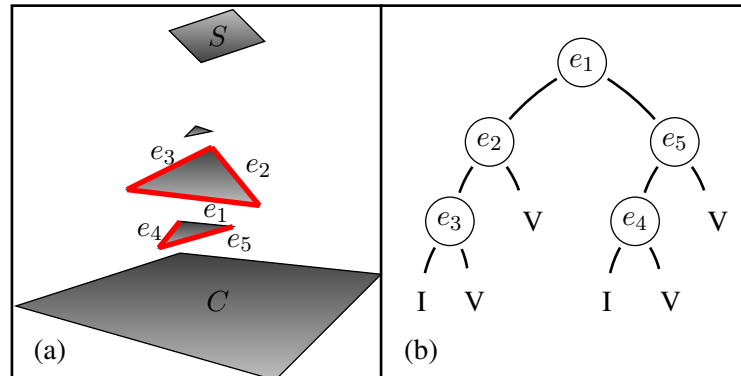


FIGURE 4.2 – Le calcul d’un arbre de visibilité prend en compte la fusion des bloqueurs. Cette fusion est effective depuis n’importe quel point du polygone S ou du polygone C .

4.1.2.1 Extraction de la visibilité en un point de vue

Les méthodes de pré-calcul d’occultation développées dans le cadre des thèses de Frédéric Mora et de Sylvain Charneau sont décrites dans le chapitre 3. Leur résultat est un arbre binaire de visibilité, dont chaque feuille contient la nature de la classe d’équivalence des droites associées : visible ou invisible. Les noeuds internes contiennent l’hyperplan associé à la droite support d’une arête d’un occulteur ayant engendré la découpe. Un exemple d’arbre de visibilité entre deux polygones S et C est donné sur la figure 4.2.

La visibilité ne dépendant que de la géométrie d’une scène, le calcul des ombres douces commence par le précalcul et la sauvegarde des arbres de visibilité entre chaque source lumineuse et chaque polygone de la scène. Le calcul de L_{direct} en un point x d’un polygone C est le suivant :

- Pour chaque source lumineuse S visible de C :
 - Prendre l’arbre de visibilité associé à S et C .
 - Extraire de l’arbre une subdivision convexe des parties visibles de S depuis x .
 - Pour chaque partie visible :
 - Calculer analytiquement sa contribution en x .
 - Sommer le résultat à l’énergie totale reçue en x .

Notre approche se distingue des précédents travaux sur plusieurs points. Le précalcul de la visibilité unifie volumes d’ombre et de pénombre au sein d’une même structure. De plus, ce calcul effectuant la fusion des bloqueurs dans \mathfrak{D}_3 , l’occultation est totalement prise en compte (cf. figure 4.2). A notre connaissance, ce n’est le cas pour aucune autre approche antérieure. Notre approche se démarque aussi des méthodes basées sur une subdivision de la scène, où un calcul précis de l’illumination est effectué uniquement en chaque sommet du maillage, permettant un calcul d’illumination par interpolation en un point quelconque. Notre méthode ne nécessite aucune subdivision, et l’illumination directe est obtenue avec le même niveau de précision en tout point de la scène.

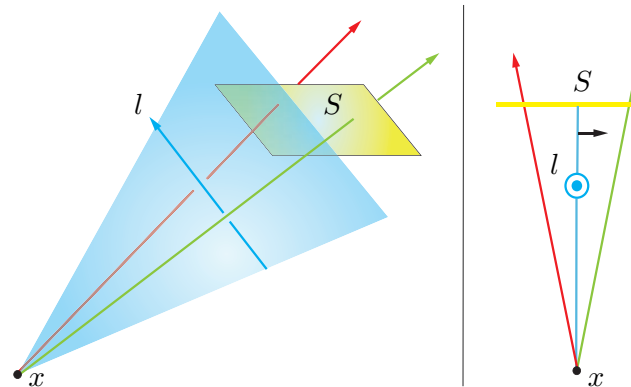


FIGURE 4.3 – Le plan défini par le point x et la droite l sépare les droites traversant x puis S en deux ensembles, ayant une orientation par rapport à l respectivement positive ou négative.

4.1.2.2 Droites passant par un point puis un polygone

Soient D_{xS} l'ensemble des droites passant par un point x puis traversant un polygone convexe S de sommets a_i . La version simplifiée du théorème du polytope minimal (cf. théorème 3.1) nous indique que ces droites sont contenues dans l'enveloppe convexe des droites xa_i . La seule connaissance de x et des sommets a_i suffit pour décrire cet ensemble.

Soit une droite l de \mathfrak{G}_3 placée entre x et la source S , et telle que x n'appartienne pas à l : depuis x , elle sépare la vue de la source en deux parties. Une première approche d'une telle séparation utilise l'opérateur *side*, qui donne l'orientation entre deux droites de \mathfrak{G}_3 (cf. section 2.3.2) : les droites dans D_{xS} sont ainsi correctement séparées par l dans l'espace des droites. Dans un souci de rapidité, nous utilisons cependant une approche complémentaire fonctionnant dans \mathfrak{G}_3 : ces mêmes droites de D_{xS} sont pareillement séparées par le plan xl contenant x et l .

La figure 4.3 illustre ce principe : la droite rouge (à gauche) a une orientation horaire par rapport à l ; à l'inverse, la droite verte (à droite) a une orientation anti-horaire ou trigonométrique. Toutes les droites dans une même classe d'équivalence par rapport à l traversent deux parties distinctes de la source S . Ces deux parties sont calculables dans \mathfrak{G}_3 , par une simple découpe de S par le plan xl formé par la droite l et le point x .

Notons que si x appartient à la droite l , alors toutes les droites passant par x puis traversant la source sont incidentes à l en x . Il suffit d'associer arbitrairement l'orientation nulle à un des deux côtés du plan xl .

4.1.2.3 Exemple d'extraction

La figure 4.4 propose un exemple d'extraction des parties visibles d'un polygone S depuis un point x sur un polygone C . L'algorithme débute à la racine de l'arbre de visibilité associé à S et C . La position de S est testée par rapport au plan défini par l'arête orientée e_1 et x

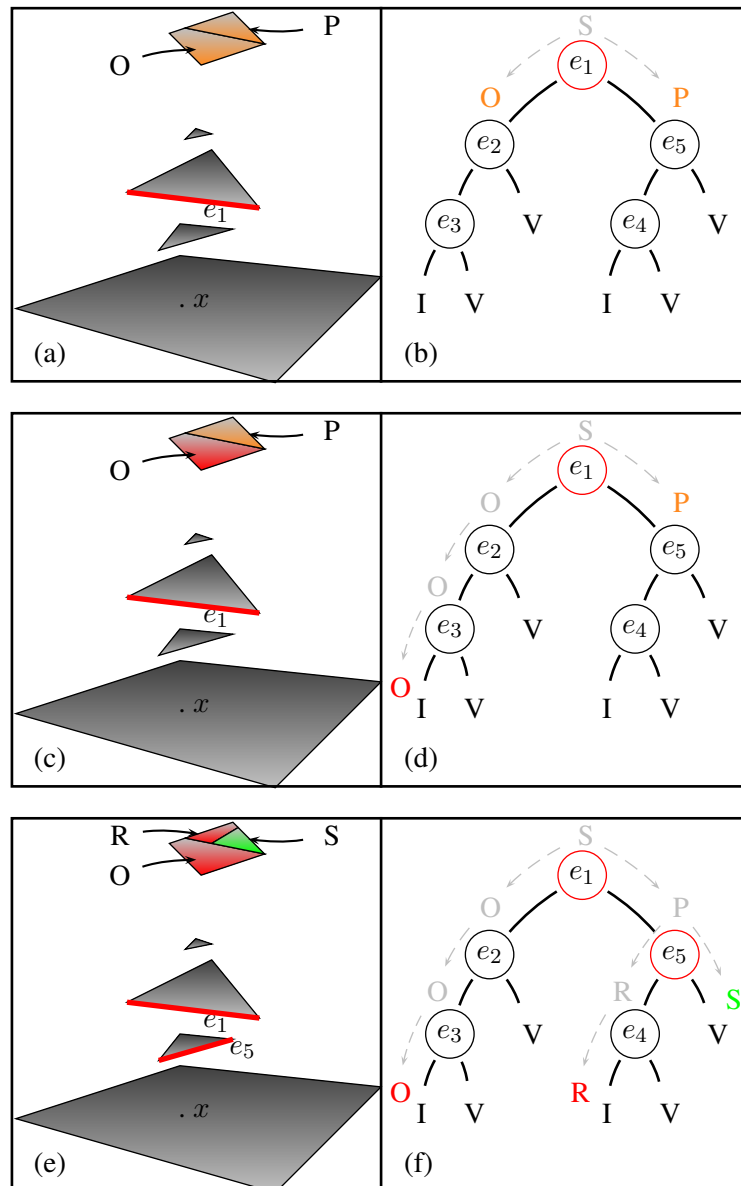


FIGURE 4.4 – Extraction de la visibilité du polygone S depuis un point x sur le polygone C

(figure 4.4 (a & b)). L'intersection de ce plan avec S produit deux fragments O et P . L'algorithme se poursuit dans les sous-arbres gauche et droit avec les fragments correspondants (figure 4.4 (c & d)). Ici, le fragment O atteint une feuille marquée invisible. Cette partie de S n'est donc pas visible depuis x (figure 4.4 (e & f)). Le fragment P est encore une fois traversé, donnant un fragment R invisible et un fragment S visible. Au final l'algorithme procure une subdivision convexe des parties de S visibles ou invisibles depuis x .

Plutôt que d'effectuer des calculs d'intersection plan/polygone en dimension 3, une optimisation consiste à projeter depuis x la droite support d'une arête e_i dans le plan du polygone

S. On se ramène ainsi à des intersections en dimension 2 de type demi-espace/polygone.

4.1.2.4 Résultats

Cet algorithme d'extraction a été utilisé dans plusieurs publications internationales [Mora 2005a, Charneau 2007a], toujours dans l'objectif d'illustrer la méthode de calcul d'occultation présentée. Dès les premiers travaux, le rendu d'une image est comparé à une méthode de référence aussi bien en terme de temps d'exécution que de qualité de rendu. Inévitablement, le rendu avec une méthode de visibilité exacte est de meilleure qualité que celui à base d'estimation de Monte-Carlo. Néanmoins, dès les premiers travaux, le temps de calcul était aussi largement réduit avec notre méthode.

Dans les travaux de F. Mora, la méthode de référence est Mental Ray 3.4¹, un *raytracer* stochastique commercial. Il est paramétré pour utiliser 256 rayons d'ombrage par point rendu. Notre méthode est de 60 à 177 fois plus rapide pour le rendu de l'ombre.

Nous avons pu constater qu'une requête de visibilité a un comportement de type logarithmique par rapport à la taille d'un arbre de visibilité. Ce comportement n'est pas étonnant, puisque le sous-ensemble des droites passant par un point de vue x donné est plus petit et surtout bien moins complexe que l'ensemble des droites entre deux polygones.

Le défaut de cette première méthode de calcul de visibilité réside en pratique dans la taille d'un arbre de visibilité. En raison d'un rejet non optimal les arbres sont trop volumineux. Avec le rejet introduit dans nos travaux avec S. Charneau, la taille des arbres est réduite. Si les temps de calcul d'une image avec ombre douce s'en trouvent quelque peu améliorés (en raison du comportement logarithmique), notons que le principal avantage réside alors dans la taille réduite des arbres, et donc dans l'économie de mémoire effectuée.

4.1.3 Méthode à base d'évaluation paresseuse

L'évaluation paresseuse d'un arbre de visibilité est décrite dans le chapitre 3, §3.4.2. Elle diffère d'un pré-calcul d'occultation principalement sur deux points :

- le test de rejet est moins performant, puisque les polytopes contenant un ensemble de droites ne sont plus calculés ;
- le calcul est effectué à la demande, lorsqu'une requête de visibilité requiert une information non encore calculée.

Cette méthode évite tout pré-calcul, sauf celui de l'ensemble des occulteurs entre la source et la cible (ce qui est rapide avec un *shaft culling* [Haines 2000]).

Nous avons appliqué cette évaluation paresseuse au traitement des ombres douces [Mora 2012, Mora 2013]. Éviter de calculer toute la visibilité dans l'espace des droites permet d'augmenter considérablement la complexité visuelle entre le polygone source et le polygone cible. En effet, le calcul d'occultation n'est jamais réalisé intégralement, ce qui est l'essence même d'une méthode paresseuse. L'information n'est calculée que lorsqu'elle

1. <http://www.mentalimages.com/>

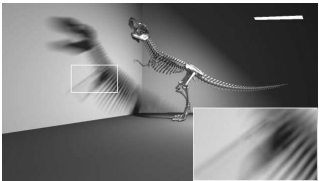


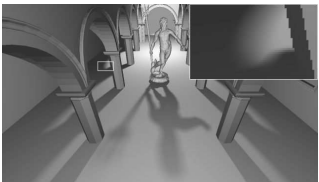




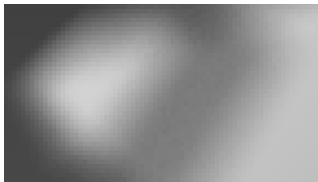
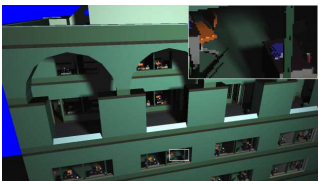


Méthode Paresseuse	MC : temps identique	MC : qualité identique
 6.5 secondes	 7 sec. / 32 rayons	 1 min. 22 sec. / 512 rayons
 7 secondes	 7 sec. / 32 rayons	 23 sec. / 256 rayons
 6 secondes	 6 sec. / 32 rayons	 24 sec. / 256 rayons
 5 secondes	 8 sec. / 32 rayons	 38 sec. / 256 rays

FIGURE 4.5 – De haut en bas : T-Rex, Sponza + Neptune, Conference et Soda Hall. La colonne de gauche montre les résultats obtenus avec notre méthode paresseuse. Les colonnes du milieu et de droite proposent un zoom sur les mêmes images calculées par une méthode de Monte-Carlo avec : au centre un temps identique, à droite une qualité similaire.

est nécessaire.

Justement, un calcul d’occultation s’intéresse à des ensembles de droites dans \mathcal{D}_n qui, potentiellement, non aucune existence réelle. Ainsi, il n’est pas rare qu’un ensemble de droites calculé compte uniquement des droites imaginaires. Utiliser un calcul à la demande permet de mettre de côté ce genre d’ensembles de droites. Une requête ne va jamais demander l’exploration de ces ensembles improbables, et donc permet d’éviter des calculs complexes et inutiles.

La méthode développée avec F. Mora (après son recrutement à l’Université de Limoges) permet ainsi d’effectuer un rendu avec ombre douce sur des scènes de test classiques, telles que “conference”, “soda hall”. Les résultats visuels sont proposés sur la figure 4.5. La taille de ces images est 1280 par 720 pixels. Les calculs sont effectués avec 4 threads, sur un Core

i7 doté de 4 Go de mémoire.

Notons que les mêmes images générées par un logiciel commercial (Mental Ray[©]) sous la même configuration matérielle et logicielle, nécessitent les temps de rendu suivants : T-Rex (512 échantillons - 4 minutes et 43 secondes) ; Sponza+Neptune (256 échantillons - 3 minutes et 52 secondes), Conference (256 échantillons - 2 minutes et 20 secondes) et Soda Hall (256 échantillons - 3 minutes et 4 secondes). Ces temps ne concernent que le calcul des ombres. Ceci montre que notre implantation du rendu avec ombres douces par méthode de Monte-Carlo est efficace.

Dans [Mora 2012], nous proposons une étude plus complète de l'influence de la taille des sources lumineuses, du nombre d'occulteurs, du nombre de sources, et de la consommation mémoire. Le principal inconvénient de cette technique réside dans cette dernière : la taille d'un arbre atteint régulièrement quelques dizaines de Mo. Si pour le rendu d'ombres, cette consommation est raisonnable, il faut savoir qu'une grande partie du temps de calcul réside en réalité dans la gestion mémoire. Notamment, la destruction d'un arbre est relativement lente, en raison d'un émiettement de la mémoire. Cela rend l'algorithme peu praticable pour un rendu interactif sur carte graphique.

4.2 Autres applications de l'évaluation paresseuse

Nous avons appliqué l'évaluation paresseuse à deux autres problématiques en rendu : la réduction des biais dans les cartes de photons, et le calcul de l'*occlusion ambiante*.

4.2.1 Réduction des biais dans les cartes de photons

Ce travail, en cours de soumission en revue internationale, a été effectué dans le cadre de la thèse de R. Perrot. L'objectif est la réduction des biais dans la méthode des cartes de photons. Nous rappelons ces biais, puis proposons une nouvelle formulation de l'estimation de densité pour le rendu, puis expliquons comment réduire ou supprimer les biais, et enfin donnons quelques résultats.

4.2.1.1 Le biais dans l'estimation de densité

L'estimation de densité [Izenman 2008] et son application aux cartes de photons sont décrits dans l'annexe C. Rappelons que le biais de l'estimation de densité s'écrit :

$$b(\hat{f}_h(x)) = f(x) \int_{-\infty}^{\infty} K_1(z) dz - f(x) - hf'(x) \int_{-\infty}^{\infty} zK_1(z) dz + \frac{h^2}{2} f''(x) \int_{-\infty}^{\infty} z^2 K_1(z) dz + o(h^2). \quad (4.2)$$

Lorsque les conditions de normalisation (cf. équation (C.4)), de symétrie (cf. équation (C.5)) et de non divergence (cf. équation (C.6)) sont respectées, alors le biais devient :

$$b(\hat{f}_h(x)) = \frac{h^2}{2} f''(x) \sigma_K^2 + o(h^2).$$

Pour minimiser le biais, il faut donc estimer une fonction f dont les variations secondaires sont faibles, et utiliser une largeur de bande h modeste.

A proximité d'un bord du domaine, cette approximation du biais n'est plus possible. Prenons un exemple : la fonction $f(x) = 1$ pour $x \in \mathbb{R}^+$ est estimée en $x < h$. La contrainte de domaine impacte directement l'expression du biais. Avec un noyau K_1 nul en tout point sauf sur l'intervalle $[-1, +1]$, il vient :

$$b(\hat{f}_h(x)) = f(x) \int_0^{x+h} K_1(z) dz - f(x) - hf'(x) \int_0^{x+h} z K_1(z) dz + \frac{h^2}{2} f''(x) \int_0^{x+h} z^2 K_1(z) dz + o(h^2). \quad (4.3)$$

Clairement les propriétés du noyau ne permettent plus, à proximité d'un bord du domaine, de réduire le biais comme précédemment : il est maintenant fonction de f .

Jones propose entre autres techniques plus efficaces de normaliser le noyau dans ce genre de configurations [Jones 1993], afin d'obtenir aux limites du domaine un biais en $hf'(x)$. Nous utilisons cette approche, qui en dimension 3 n'implique pas trop de calculs (contrairement aux autres méthodes proposées par Jones).

4.2.1.2 Le biais dans les cartes de photons

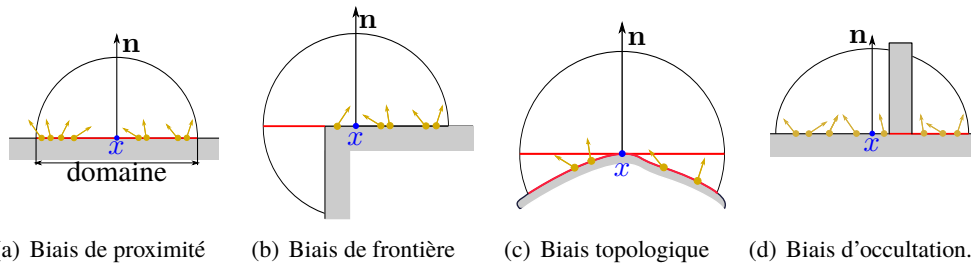


FIGURE 4.6 – Classification des biais pour l'estimation de densité. Les observations (les photons) sont en jaune, le point d'estimation x en bleu, les sources du biais en rouge.

Plusieurs taxonomies des biais dans les cartes de photons ont été proposées [Schregle 2003, Tobler 2006] ; nous distinguons les quatre biais suivants :

- Biais de proximité (figure 4.6(a)) : intrinsèque à l'estimation de densité, il est dû aux variations de la fonction f .
- Biais de frontière (figure 4.6(b)) : apparaît aux bords des objets lorsque la normalisation du noyau (cf. équation (C.4)) n'est plus respectée.

- Biais topologique (figure 4.6(d)) : des photons sont (mal) projetés sur un domaine de dimension 2, conduisant à considérer un domaine d'estimation trop grand.
- Biais d'occlusion (figure 4.6(d)) : des photons non visibles du point d'observation sont utilisés, ou au contraire les ancêtres des photons ne voient pas le point d'estimation.

Si le premier biais ne peut pas être corrigé, étant intrinsèque à l'estimation de densité, les autres peuvent être réduits voire totalement supprimés. Depuis l'introduction des cartes de photons, beaucoup de travaux ont essayé de traiter ces biais, généralement par des méthodes algorithmiques [Schregle 2003, Lavignotte 2002, Jensen 2001, Tobler 2006, Lavignotte 2003, Lastra 2002, Havran 2005a, Bekaert 2003, Chen 2010].

Notre idée est que la visibilité (*i.e.* un calcul d'occultation) doit permettre de calculer correctement les domaines d'estimation, rendant possible le respect de la propriété C.4 de normalisation du noyau. Dans ce but, nous devons faire apparaître la visibilité dans l'estimation de densité utilisée dans les cartes de photons.

4.2.1.3 Nouvelle formulation pour le rendu

Notre méthode repose sur une nouvelle formulation de l'estimation de densité dans les cartes de photons. Nous utilisons la propriété de conservation de la luminance le long d'un rayon (*cf.* §A.1.3.1). Ainsi, plutôt que de calculer la luminance partant d'un point d'estimation x , nous exprimons celle reçue en un point d'observation y qui *regarde* x à condition que x et y soient en visibilité mutuelle :

$$L_i(y, x \rightarrow y) = L_o(x, x \rightarrow y)V(x \leftrightarrow y).$$

L'estimation de densité sur $L_i(y)$ est alors :

$$L_i(y, x \rightarrow y) \approx \frac{1}{n|\mathbf{H}|} \sum_{i=1}^n K_3(\mathbf{H}^{-1}(x - x_i)) L_o(x_i, x_i \rightarrow y)V(x_i \leftrightarrow y),$$

ce qui nous permet de faire apparaître un premier terme de visibilité. La luminance $L_o(x_i)$ correspond à chaque photon i . Elle peut être remplacée en utilisant l'ETL, ce qui conduit à introduire un second terme de visibilité entre x et les ancêtres x_i^{-1} de chaque photon x_i :

$$L_i(y, x \rightarrow y) \approx \frac{1}{n|\mathbf{H}|} \sum_{i=1}^n f_r(x_i, x_i \rightarrow x_i^{-1}, x_i \rightarrow y) K_3(\mathbf{H}^{-1}(x - x_i)) L_o(x_i^{-1}, x_i^{-1} \rightarrow x_i) G(x, x_i^{-1}) V(x_i \leftrightarrow y).$$

En utilisant l'énergie transportée α_i du photon i , il vient :

$$L_i(y, x \rightarrow y) \approx \frac{1}{n|\mathbf{H}|} \sum_{i=1}^n f_r(x, x_i \rightarrow x_i^{-1}, x_i \rightarrow y) K_3(\mathbf{H}^{-1}(x - x_i)) \alpha_i V(x \leftrightarrow x_i^{-1}) V(x_i \leftrightarrow y). \quad (4.4)$$

Introduire ainsi deux termes de visibilité nous permet de réduire voire supprimer les biais topologique, d'occlusion et de frontière, comme discuté dans les paragraphes suivants.

Le biais de frontière est un problème de visibilité : le domaine d'estimation est défini généralement par le seul noyau (un disque placé sur le point d'estimation), et moins souvent par la seule géométrie [Tobler 2006]. Néanmoins, si elle est nécessaire pour obtenir un domaine d'estimation correct, la prise en compte de toute la géométrie vue depuis l'observateur y ne suffit pas pour régler le biais de frontière. Une fois le domaine connu, il faut encore respecter l'équation (C.4).

Ainsi, un bonne solution à ce problème doit :

- calculer la géométrie visible point à surface entre l'observateur y et le voisinage de x ;
- normaliser le noyau sur cette géométrie.

Nous proposons deux solutions à ces problèmes :

- Utiliser le calcul d'occultation à la demande pour régler le premier, sachant que la réutilisation des requêtes doit permettre de rentabiliser les calculs [Mora 2012].
- La normalisation du noyau sur une géométrie connue passe par la possibilité d'intégrer le noyau sur des polygones quelconques. L'utilisation d'un noyau produit permet de se limiter à des polygones à arêtes droites (et non par courbes, ce qui est le cas en utilisant un noyau radial sphérique), autorisant des résultats numériques précis.

En clair, notre solution utilise uniquement des noyaux produits normalisés sur un ensemble de polygones visibles depuis un point de vue donné. Utiliser un noyau produit n'est pas, en soit, un problème. Si habituellement un noyau symétrique radial est privilégié, un noyau produit possède un comportement similaire (cf. figure 4.7).

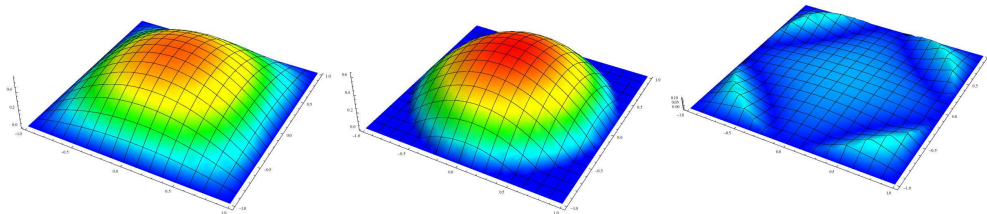


FIGURE 4.7 – Différence entre deux noyaux en dimension 2 : (a) noyau produit d'Epanechnikov, (b) noyau symétrique radial d'Epanechnikov, (c) différence entre les deux..

4.2.1.4 Réduire le biais de frontière

4.2.1.5 Supprimer le biais d'occlusion

Le biais d'occlusion est intrinsèquement un problème de visibilité. Notre nouvelle formulation (cf. équation (4.4)) fait apparaître explicitement deux fois le terme de visibilité V . Par conséquent, traiter le biais d'occlusion nécessite deux calculs de visibilité :

- un premier entre les positions x_i de chaque photon i et le point d'observation y ;
- un second entre la position d'estimation x et les ancêtres x_i^{-1} de chaque photon i .

Les calculs de visibilité peuvent être effectués point à point ou par groupe de points, en fonction de la méthode de visibilité utilisée. Le calcul d'occultation utilisé pour réduire le

biais de frontière permet aussi de déterminer rapidement la visibilité point à point, à moindre coût.

Notons que le traitement du biais d’occlusion renforce le biais de frontière. Le traitement du premier nécessite ainsi celui du second. Nous avons deux approches pour le traitement du biais de frontière :

- une normalisation classique du domaine en x avec une visibilité depuis y ;
- une normalisation par photon, avec un domaine restreint à la géométrie vue depuis y et depuis l’ancêtre x_i^{-1} .

La seconde approche, qui est la seule “juste”, est malheureusement excessivement gourmande en terme de ressources de calcul. Elle est donc à utiliser avec prudence.

4.2.1.6 Supprimer le biais topologique

Le biais topologique est engendré par une mauvaise projection des photons dans \mathcal{G}_3 sur un noyau de domaine d’estimation de dimension 2. Si beaucoup d’auteurs utilisent un noyau de dimension 2, c’est avant tout pour réduire les problèmes de biais, notamment d’occlusion.

En traitant les biais globalement, une telle simplification devient inutile. Ainsi, ce problème disparaît naturellement avec un noyau de dimension 3.

4.2.1.7 Résultats

Nous avons testé notre méthode dans trois algorithmes de rendu reposant sur les cartes de photons : la visualisation directe, la visualisation indirecte et les cartes de photons progressives.

La carte de photons est générée classiquement, seule l’estimation de densité est modifiée. Nous utilisons l’évaluation paresseuse pour les calculs de visibilité, ou une méthode de type Monte-Carlo à des fins de comparaisons. Le noyau est de type Epanechnikov produit, en dimension 3. La largeur de bande et le nombre n de photons utilisés dans l’estimation sont des paramètres de l’utilisateur.

Nos résultats confirment la réduction des biais, visibles même avec une visualisation indirecte. La figure 4.8 illustre les différents résultats sur une scène volontairement simple, en visualisation directe. La figure 4.9 montre le résultat du traitement des biais topologique et de frontière sur la scène Sibenik. Les différences sont évidemment aux bords des polygones.

Le traitement des biais en visualisation indirecte est illustré sur la figure 4.10. La carte de photons utilisée est exactement la même que dans la figure 4.8. Même en visualisation indirecte, les biais d’occlusion sont présents. La figure montre bien que la correction de l’occlusion nécessite aussi le traitement du biais de frontière.

Enfin notons que les temps de calculs sont raisonnables, sauf lorsque le biais d’occlusion est traité en totalité. La table 4.1 donne les temps de calcul du facteur de normalisation pour le traitement du biais de frontière seul, ou pour le traitement des deux biais de frontière et d’occlusion (le biais topologique est toujours corrigé avec un noyau en dimension 3). Les

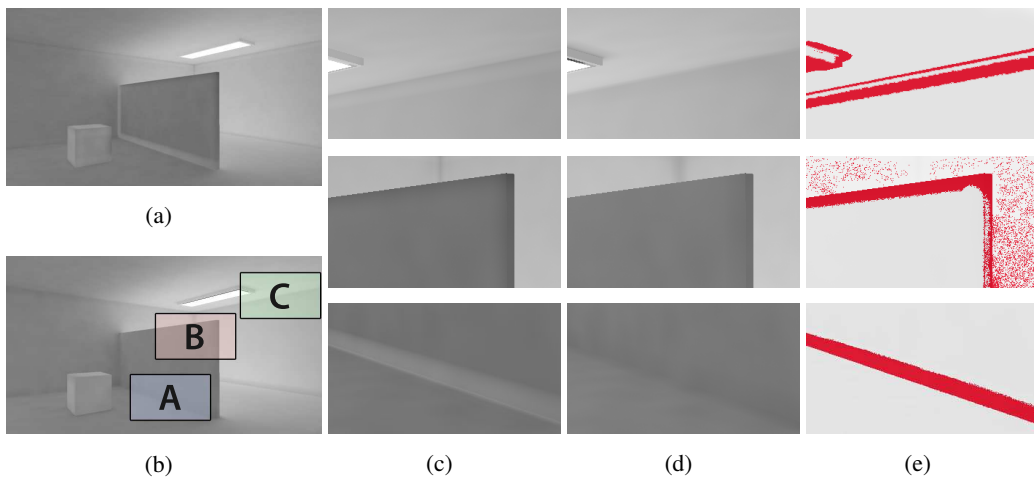


FIGURE 4.8 – Visualisation directe : (a) classique ; (b) avec corrections ; (c) zoom classique ; (d) zoom avec corrections ; (e) différences entre (d) et (c). Biais dans les lignes : occlusion en bas (A), frontière au milieu (B) et topologique en haut (C).

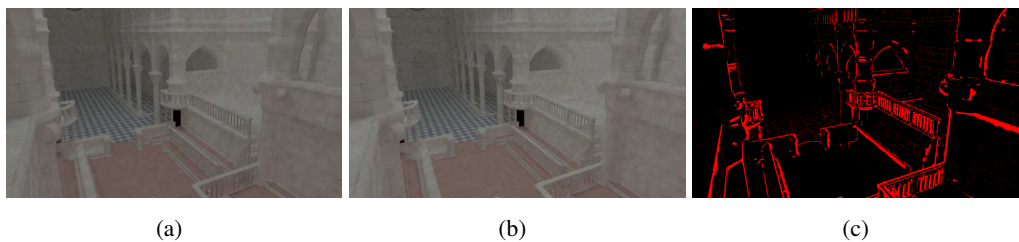
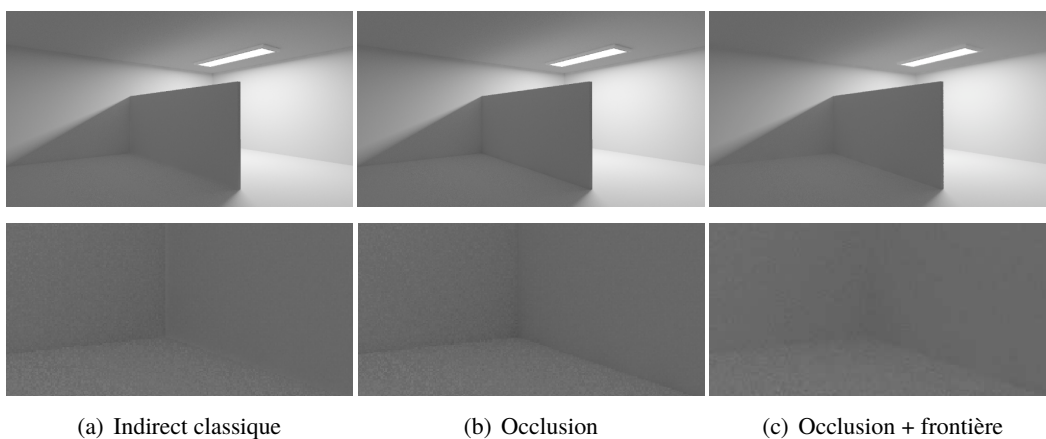


FIGURE 4.9 – Exemple de correction des biais de frontière et topologique : (c) montre la différence entre les images (a) sans correction et (b) avec corrections.



(a) Indirect classique (b) Occlusion (c) Occlusion + frontière

FIGURE 4.10 – Exemples de rendu en visualisation indirecte (zoom en bas).

colonnes “NO-MC” et “NO-Ex” montrent qu’un calcul d’occultation paresseux est avantageux en terme de temps de calculs par rapport à une approximation de type Monte-Carlo ; en

Scène	NO-MC	NO-Ex	NOP-MC	NOP-Ex
C. Box	10s	7s	2h24m	57m
Half	4m 3s	2m 1s	35h 25m	14h 45m
Shutter	1m 19s	27s	1h 11m	12m 51s
Class.	6m 29s	1m 40s	7h 40m	7h 5m
Sponza	5m 15s	2m 23s	18h 43m	63h 19m

TABLE 4.1 – Temps de calculs pour le rendu avec normalisation. NO : depuis l’observateur y . NOP : depuis y et les ancêtres des photons x_i^{-1} . Ex : calcul analytique exact. MC : calcul approché par du Monte-Carlo.

effet, la ré-exploitation des calculs développés dans les requêtes antérieures permet d’amortir les premiers développements de l’arbre, et ainsi de gagner beaucoup de temps. A l’inverse, en normalisant les noyaux depuis les ancêtres de chaque photon, la ré-exploitation n’est plus possible (trop d’arbres dans le cache LRU), car les arbres non utilisés depuis un certain temps sont supprimés pour faire de la place en mémoire. Ceci est directement en lien avec une trop grande taille des arbres de visibilité.

4.2.2 Occlusion ambiante

Nous présentons ici l’application de nos travaux sur l’occlusion ambiante, une approximation géométrique de l’illumination globale [Apostu 2012]. Cette étude fût conduite durant la thèse d’Oana Apostu, encadrée par F. Mora et D. Ghazanfarpour de l’Université de Limoges.

4.2.2.1 Problématique et Travaux précédents

L’occlusion ambiante (OA dans la suite) est un modèle d’illumination globale empirique, à faible coût. Zhukov *et al.* introduisent une version plus complexe, l’*obscurance* [Zhukov 1998], adaptée ensuite par l’industrie du cinéma [Landis 2002, Bredow 2002]. Un état de l’art sur les méthodes d’*obscurance* et d’ OA est disponible dans [Méndez-Feliu 2009]. En résumé, l’ OA est une propriété géométrique d’un point approchant le pourcentage de luminance incidente par la géométrie visible proche. Son expression est :

$$OA(x) = \frac{1}{\pi} \int_{\omega \in \Omega} Vis_{\delta}(x, \omega) (\mathbf{n} \cdot \omega) d\omega, \quad (4.5)$$

où x est le point recevant l’occultation ; \mathbf{n} la normale au point x ; Ω est l’hémisphère supérieur de direction principale \mathbf{n} ; Vis_{δ} est une fonction de visibilité valant 0 lorsqu’aucune géométrie n’est visible depuis x dans la direction ω et pour une distance maximale δ , et 1 sinon (dans la version initiale de l’*obscurance*, cette fonction retourne une atténuation dépendant de la distance au point d’intersection).

Baucoup de travaux précédents essaient de calculer efficacement et correctement l’ OA : les méthodes à base de lancé de rayons [Méndez 2003, Méndez 2004], qui nous serviront de

référence dans la suite de cette section ; les méthodes analytiques telles que [Bunnell 2005, Hoberock 2007] qui souffrent d'artefacts visuels, ou telle que [McGuire 2010] dont la visibilité approchée engendre des problèmes de sur-occultation, ou encore [Laine 2010] via un échantillonnage des données utilisant notamment des niveaux de détails pour simplifier la visibilité lointaine, et enfin [Szirmay-Kalos 2010] dont la version volumétrique de l'OA fonctionne très rapidement sur GPU, au prix d'une sensibilité au bruit et donc de quelques imprécisions ; enfin des méthodes dédiées à un contexte d'utilisation précis [Tarini 2006, Hegeman 2006, Kontkanen 2006, Kirk 2007], contenant ainsi des approximations les rendant inutilisables en toute généralité ; notons cependant dans le domaine des jeux vidéo les méthodes travaillant dans l'espace écran [Mittring 2007], particulièrement rapides aux dépens de la précision du résultat.

Les méthodes sont donc soit approchées [Méndez 2003] et donc bruitées, soit analytiques [Bunnell 2005, Hoberock 2007, McGuire 2010], mais avec un calcul de visibilité approché engendrant des artefacts visuels. Notre démarche est analytique, avec une visibilité exacte afin de s'affranchir totalement de toutes erreurs visuelles.

4.2.2.2 Solution analytique de l'intégrale d'OA

Notre méthode suppose que toutes les surfaces (les polygones) sont planes et orientées. Soit x un point sur la surface \mathbf{S} de normale \mathbf{n} , et soit \mathbf{T} un polygone convexe complètement visible depuis x . Le calcul de l'OA engendrée par \mathbf{T} est équivalent à celui du facteur de forme entre x et \mathbf{T} [Zhukov 1998]. Dans le cas simple d'un polygone directement visible depuis x , le facteur de forme peut être évalué avec la formule de Lambert [Lambert 1760]. Pour \mathbf{T} , l'intégrale de l'OA (cf. équation (4.5)) devient la somme suivante :

$$OA(x, \mathbf{T}) = \frac{1}{2\pi} \sum_{i=1}^N \left(\cos^{-1} \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_{i+1}\|} \right) \left(\frac{\mathbf{v}_i \times \mathbf{v}_{i+1}}{\|\mathbf{v}_i \times \mathbf{v}_{i+1}\|} \cdot \mathbf{n} \right), \quad (4.6)$$

où N est le nombre de sommets de \mathbf{T} , et $\mathbf{v}_i = t_i - x$ sont les vecteurs liant le point x et chaque sommet t_i de \mathbf{T} . Dans le cas où plusieurs polygones sont visibles, l'OA totale est la somme des intégrales pour chacun.

L'équation (4.6) n'est valide qu'à condition que toutes les portions visibles de la géométrie soient connues. Calculer la visibilité en chaque point x nécessitant un calcul d'OA est considéré dans les travaux antérieurs comme trop coûteux. Nous observons que des points voisins d'une même surface ont des vues très proches de cette géométrie voisine. Nous utilisons donc, naturellement, un calcul de visibilité proche de nos travaux sur l'occultation : pour chaque surface, nous définissons un arbre de visibilité, qui sera ensuite utilisé en chaque point nécessitant un calcul d'OA. Un arbre de visibilité est une version modifiée d'un arbre d'occultation : la notion de profondeur est en effet nécessaire, pour calculer l'obscurité et tenir compte de la distance d'occultation maximale δ .

4.2.2.3 Nouvel arbre de visibilité

Notre arbre de visibilité utilise les deux hypothèses suivantes : les polygones sont orientés, et surtout ne se croisent pas. Lorsque cette dernière condition n'est pas respectée, il suffit de découper les deux polygones par la droite formée par leurs plans supports.

Notre arbre est basée sur la notion de classe d'équivalence de droites au sens de Pellegrini [Pellegrini 1991, Pellegrini 2004], étendue en prenant en compte notre contrainte de distance. Nous considérons ici uniquement les droites traversant une face S et sortant d'un côté donné (*i.e.* la face est orientée), et en se limitant à la géométrie proche (au plus un certain δ). Notre définition est la suivante : une cellule de la partition des droites dans \mathcal{D}_3 correspond soit à un ensemble de droites ne traversant aucun obstacle, soit un ensemble de droites traversant un premier obstacle.

Ainsi, cette nouvelle définition diffère de l'originale uniquement par l'introduction d'une notion d'ordre sur les faces traversées par l'ensemble de droites. Il est intéressant de noter que cette définition est consistante en tous points x de la surface S : notre hypothèse de géométrie sans intersection nous assure que, vue depuis S , la liste des faces traversées par les droites d'une classe d'équivalence est ordonnable. Il existe donc bien, pour une classe d'équivalence traversant au moins un obstacle, une et une seule première face traversée.

Notre nouvel arbre de visibilité ressemble beaucoup au précédent : il est binaire, chaque noeud interne contient un hyperplan dans \mathcal{D}_3 correspondant à une arête d'un occulteur, une feuille correspondant à une classe d'équivalence "vide" est notée par une simple marque. Néanmoins il possède une différence importante : une feuille "invisible" est remplacée par une feuille contenant la face correspondant à l'obstacle le plus proche depuis la surface S .

Cet arbre n'est pas pré-calculé, car la complexité d'un arrangement dans l'espace des droites \mathcal{G}_3 est toujours de l'ordre de $\mathcal{O}(n^4 \log n)$ [Pellegrini 1993]. Ainsi, nous utilisons encore une fois une évaluation paresseuse, afin de permettre son exploitation depuis un point x .

4.2.2.4 Calculer l'occlusion ambiante et l'obscurance

Le calcul de l' OA est très proche de celui réalisé pour les ombres douces. Pour l'effectuer, il est nécessaire de calculer la géométrie visible depuis un point x donné de S . Cela revient à déterminer les classes d'équivalence des rayons passant par x . Une illustration d'un tel calcul est proposée dans la figure 4.11.

L'arbre de visibilité est initialisé avec toute la géométrie visible depuis la surface orientée S , jusqu'à une certaine distance maximale δ . Cette dernière est un paramètre utilisateur, limitant le calcul de l' OA à un certain voisinage (comme dans toutes les méthodes d' OA).

Pour un point x de S , une requête utilise l'arbre de visibilité et les polygones P_i représentant une triangulation de la demi-sphère centrée en x , de rayon δ . Ces polygones sont en quelque sorte les *cibles* d'un calcul d'occultation traditionnel. Ils ne sont pas des obstacles, mais uniquement des guides pour la requête : ils définissent avec x les ensembles de droites dont on souhaite connaître les premiers obstacles rencontrés, *i.e.* de potentielles classes

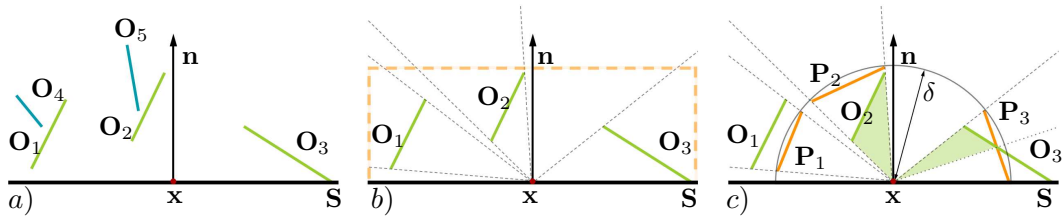


FIGURE 4.11 – Calcul de visibilité en un point x . **a)** Les obstacles O_i , pour $i \in [1 \dots 5]$, sont potentiellement visibles depuis S . En x , O_4 et O_5 ne sont pas visibles. **b)** L'hémicube centré en x sert à guider la requête de visibilité. **c)** Les occulteurs O_1 , O_2 et O_3 sont entièrement visibles depuis x , et ont respectivement la même occlusion que les fragments P_1 , P_2 et P_3 . δ est la distance d'occlusion maximale, au delà de laquelle un obstacle n'influence plus l'occlusion. O_1 et une portion de O_3 sont donc rejetés.

d'équivalence. Ils ne sont donc pas injectés dans l'arbre.

L'extraction de la visibilité consiste à filtrer l'ensemble des polygones cibles dans l'arbre de visibilité. Dans un noeud non encore visité et contenant une liste (O_i) de plus d'un occulteur, l'algorithme fonctionne comme pour le calcul des ombres douces : il choisit un occulteur O_k au hasard, remplace le noeud par l'arbre associé à O_i , et filtre les autres occulteurs.

La différence principale est que les feuilles invisibles n'existent plus : représentant des classes d'équivalence avec au moins un obstacle, elles sont remplacées par des feuilles contenant l'occulteur le plus proche de la surface. Ainsi, ces nouveaux arbres de visibilité sont-ils plus gros que les arbres d'occultation : dans les seconds, chaque noeud correspondant à la classe d'équivalence des droites traversant un occulteur devient une feuille invisible, alors que pour les premiers ils contiennent potentiellement un sous-arbre, s'il reste de la géométrie devant (bien sûr, les occulteurs se trouvant derrière O_k sont éliminés).

Le choix d'un occulteur au hasard peut sembler, dès lors, être une mauvaise idée. Pourtant, utiliser l'occulteur le plus proche (lorsqu'il est connu), a tendance à générer des arbres moins performants : cela ne permet pas d'obtenir un partitionnement de \mathcal{D}_3 qui soit optimal, comme nos expérimentations l'ont montré. Un algorithme de Las Vegas² se révèle bien meilleur.

Lorsque l'algorithme arrive sur une feuille contenant O_p , trois cas de figures se présentent :

1. O_p est à une distance moindre de δ de x , auquel cas il produit une OA , calculée avec l'équation (4.6).
2. O_p est trop éloigné ; en utilisant l'approximation de l'hémisphère centrée en x et de rayon δ , O_p est derrière un des polygones P_i .
3. une partie de O_p est visible à une distance inférieure à δ , l'autre étant trop distante.

2. Algorithme reposant sur un ou des choix aléatoires et produisant néanmoins le bon résultat ; par opposition à Monte-Carlo, qui retourne un résultat approché. Un exemple est le tri rapide avec pivot aléatoire.

Chaque polygone P_i étant utilisé comme cible, ce dernier cas de figure revient à calculer la partie de O_p devant un unique P_i donné, calcul trivial s'il en est.

Notre méthode fonctionne aussi pour calculer l'obscurance, en utilisant une approximation de l'atténuation dépendant de la distance d'occultation selon ω . La distance utilisée est constante pour une classe d'équivalence donnée, et correspond à la distance entre x et le barycentre du fragment visible de O_p .

4.2.2.5 Résultats

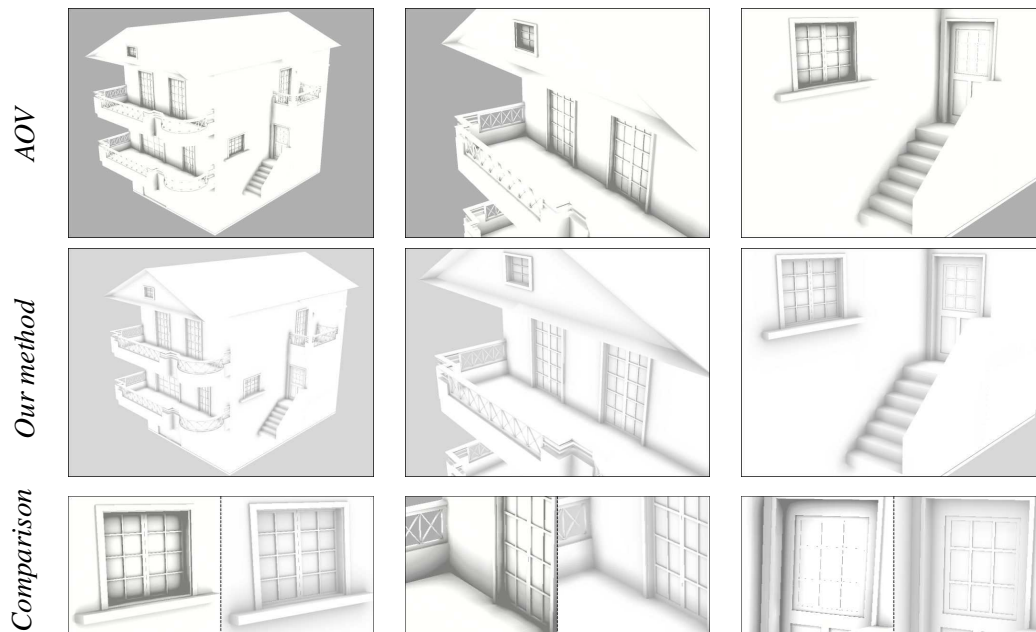


FIGURE 4.12 – Exemple de résultat. La ligne du haut concernant la méthode *Ambient Occlusion Volumes* [McGuire 2010], très rapide mais qui tend à surestimer l'occultation. La ligne du milieu est produite avec notre algorithme. La dernière ligne souligne les différences.

Les résultats sont détaillés dans [Apostu 2012]. Sur la scène illustrée dans figure 4.12, le rendu est effectué en 221 secondes avec Mental Ray[©], 27 secondes avec notre algorithme, et 0.074 secondes avec l'AOV. Si les temps de calcul peuvent paraître en défaveur de notre méthode, nous observons cependant que la qualité est largement améliorée avec notre méthode, que ce soit par rapport à Mental Ray[©] et sa technique stochastique, ou l'AOV.

L'AOV est beaucoup plus rapide, mais reste une méthode très imprécise. Pour un résultat de qualité, nous sommes à notre tour bien plus rapide que les méthodes stochastiques, comme par exemple Mental Ray[©].

En conclusion, une évaluation paresseuse de la visibilité, dérivée de celle d'occultation, nous permet de récupérer l'ensemble des surfaces visibles d'un point, et de calculer en un temps raisonnable l'occlusion ambiante ou l'obscurance à un coût bien inférieur aux méthodes précédentes, pour une qualité égale sinon supérieure. Il s'agit de la seule méthode

de calcul de l' OA qui soit à la fois analytique et exacte, à l'approximation de l'hémisphère prêt.

4.3 Autres travaux en informatique graphique

Nous proposons ici deux travaux ne reposant pas sur nos calculs d'occultation (*cf.* chapitre 3). Les premiers concernent des lobes de cosinus, permettant une application de rendu interactive. Les seconds concernent le rendu interactif en environnement architectural.

4.3.1 Lobes de cosinus

Ces travaux ont démarré avec la thèse de S. Meunier, qui a malheureusement renoncé pour raisons personnelles après deux ans de travail. Ils se sont conclus dans le cadre du doctorat de Romuald Perrot [Meunier 2010].

Les méthodes de type pré-calcul de transfert énergétique (PTE) [Sloan 2002] transforment, grâce aux *harmoniques sphériques*, l'intégrale de l'ETL en un produit de coefficients, rendant très rapide le rendu d'une image tout particulièrement sur carte graphique (*cf.* annexe D). Cependant, le pré-calcul nécessaire à leur mise en œuvre est très coûteux en temps et en mémoire. Même si des solutions ont été proposées pour diminuer la quantité de données à stocker, l'utilisation d'environnements hautes fréquences ou dynamiques reste problématique.

L'objectif de ces travaux est de fournir une représentation permettant d'utiliser des environnements dynamiques et hautes fréquences, qui soit à la fois simple et rapide à utiliser. Nous utilisons les fonctions *lobes de cosinus* [Fyffe 2009] pour homogénéiser les termes de l'ETL. Introduites pour modéliser des FDRB [Lafortune 1997], nous les généralisons à l'ensemble de l'ETL. Nous proposons de remplacer chacun des termes de cette équation par des sommes de fonctions lobes de cosinus, puis de réécrire l'ETL comme une somme d'intégrales de lobes de cosinus.

4.3.1.1 Calculer avec des lobes de cosinus

Nous présentons ici les fonctions lobes de cosinus, leur produit et intégration. Une fonction lobe de cosinus pour une direction unitaire \mathbf{o} donnée est :

$$\begin{aligned} \mathcal{C}_{(s,\mathbf{a},e)}(\mathbf{o}) &= s \max(\mathbf{a} \cdot \mathbf{o}, 0)^e \\ &= s \bar{\mathcal{C}}_{(\mathbf{a},e)}(\mathbf{o}) \end{aligned}$$

où s est un facteur d'échelle, \mathbf{a} est l'axe unitaire du lobe et e est un facteur contrôlant la largeur du lobe.

Les fonctions lobes de cosinus possèdent les propriétés suivantes :

- elles sont symétriques par rapport à l'axe du lobe. En effet, si \mathbf{o} est un vecteur et \mathbf{o}' est le symétrique de \mathbf{o} par rapport à l'axe \mathbf{a} alors $\mathcal{C}_{(s,\mathbf{a},e)}(\mathbf{o}) = \mathcal{C}_{(s,\mathbf{a},e)}(\mathbf{o}')$;

- projetées sur n'importe quel plan contenant l'axe \mathbf{a} , elles s'expriment comme des fonctions de l'angle entre \mathbf{a} et \mathbf{o} ;
- dans un tel plan, elles sont croissantes monotones puis décroissantes monotones. Le point d'inflexion entre ces deux courbes se situe pour un angle nul, soit $\mathbf{o} = \mathbf{a}$.

Les sommes de lobes de cosinus construisent des fonctions évoluées :

$$\mathcal{S}(\mathbf{o}) = \sum_i \mathcal{C}_{(s_i, \mathbf{a}_i, e_i)}(\mathbf{o}) = \sum_i \mathcal{C}_i(\mathbf{o}).$$

Néanmoins, et comme les fonctions à base radiale sphérique, les lobes de cosinus ne forment pas une base orthonormale. L'intégrale d'un lobe est donnée par :

$$I = \int_{\Omega} \mathcal{C}_{(s_1, \mathbf{a}_1, e_1)}(\mathbf{v}) \, d\mathbf{v}.$$

L'axe \mathbf{a}_1 n'a aucune incidence dans cette intégration, un lobe étant une fonction radiale à base sphérique. L'intégrale I se simplifie alors en :

$$I = s_1 \int_{\Omega_{\mathbf{a}_k}} (\mathbf{a}_k \cdot \mathbf{v})^{e_1} \, d\mathbf{v}.$$

L'intégrale d'un lobe ne dépend donc que de son exposant. Cela nous permet de pré-calculer les valeurs de I pour un grand nombre d'exposants. L'intégrale a une valeur d'autant plus faible que l'exposant est fort. Notre tabulation privilégie donc les faibles exposants (comportement d'indice logarithmique).

Le produit de deux lobes de cosinus est approché par :

$$\mathcal{C}_{(s_1 s_2 p_r, \mathbf{a}_r, e_r)}(\mathbf{o}) \approx \mathcal{C}_{(s_1, \mathbf{a}_1, e_1)}(\mathbf{o}) \mathcal{C}_{(s_2, \mathbf{a}_2, e_2)}(\mathbf{o}) \quad (4.7)$$

où p_r est appelé *facteur d'échelle partiel*. Il s'agit bien d'une approximation, comportant une légère erreur. Nous calculons un tel lobe de façon à minimiser cette erreur, soit :

$$\{\mathbf{a}_r, p_r, e_r\} = \arg \min_{\{\mathbf{a}_r, p_r, e_r\}} \int_{\Omega} (\mathcal{C}_1(\mathbf{o}) \mathcal{C}_2(\mathbf{o}) - \mathcal{C}_r(\mathbf{o}))^2 \, d\mathbf{o}.$$

Nous avons montré que le lobe produit a toujours un exposant supérieur ou égal au plus grand des exposants des deux opérandes. Cette propriété est particulièrement utile dans la mesure où l'erreur commise dans le calcul du lobe produit sera d'autant plus faible que ce dernier possède un exposant élevé.

Le produit de deux lobes est pré-calculé dans une table, en privilégiant les exposants faibles pour minimiser les erreurs de calculs lors de l'intégration d'un lobe produit.

4.3.1.2 Application au rendu

Nous transformons l'équation de transfert lumineux en l'exprimant à l'aide de lobes de cosinus uniquement. Ainsi, nous partons de l'équation (A.7). Le terme émissif L_e étant

généralement constant, nous le supprimons de la discussion. Dans l'intégrale décrivant la luminance réfléchie, nous fusionnons la FDRB et $(\mathbf{n}_x \cdot \mathbf{i})$. Il vient :

$$L_o(x, \mathbf{o}) = L_{direct} = \int_{\Omega_s} L_s(x, \mathbf{i}) \tilde{f}_r(x, \mathbf{i}, \mathbf{o}) V(x, \mathbf{i}) d\omega_{\mathbf{i}}$$

où $L_s(x, \mathbf{i})$ est une contribution directe dans la direction $-\mathbf{i}$. Cette équation fait apparaître trois termes : la luminance directe L_s , la FDRB (modifiée) \tilde{f}_r et la visibilité V . Notre objectif est d'exprimer chacun de ces termes de façon homogène selon une somme de lobes. Après transformations nous obtenons (pour x et \mathbf{o} fixés) :

$$\tilde{f}_r(\mathbf{i}) \approx \sum_i C_i^{\tilde{f}}(\mathbf{i}), \quad L_s(\mathbf{i}) \approx \sum_j C_j^L(\mathbf{i}), \quad V(\mathbf{i}) \approx \sum_k C_k^V(\mathbf{i}).$$

Nous pouvons réécrire l'ETL de la façon suivante :

$$L_o(x, \mathbf{o}) \approx \int_{\Omega_x} \sum_j C_j^L(\mathbf{i}) \sum_i C_i^{\tilde{f}}(\mathbf{i}) \sum_k C_k^V(\mathbf{i}) d\omega_{\mathbf{i}}. \quad (4.8)$$

Une méthode de type PTE utilise des termes pré-calculés permettant d'approcher L_o par une somme d'intégrales pondérées, interdisant les environnements dynamiques. Au contraire, reposant sur des pré-calculs ne dépendant pas des scènes, notre approche est exploitable dynamiquement.

Nous exploitons les propriétés du produit pour ré-écrire l'ETL sous la forme suivante :

$$L_o \approx \int_{\Omega_x} \sum_{ij} \left[s_i^{\tilde{f}} \bar{C}_i^{\tilde{f}}(\mathbf{i}) s_j^L \bar{C}_j^L(\mathbf{i}) \right] \sum_k (s_k^V \bar{C}_k^V(\mathbf{i})) d\omega_{\mathbf{i}}. \quad (4.9)$$

En utilisant l'approximation (4.7), le premier produit de deux lobes est approché par :

$$s_m \bar{C}_m(\mathbf{i}) \approx s_i^{\tilde{f}} \bar{C}_i^{\tilde{f}}(\mathbf{i}) s_j^L \bar{C}_j^L(\mathbf{i})$$

L'équation (4.9) devient :

$$L_o(x, \mathbf{o}) \approx \int_{\Omega_x} \sum_m s_m \bar{C}_m(\mathbf{i}) \sum_k s_k^V \bar{C}_k^V(\mathbf{i}) d\omega_{\mathbf{i}}. \quad (4.10)$$

En répétant cette démarche sur les deux derniers lobes, il vient :

$$L_o(x, \mathbf{o}) \approx \sum_p s_p \int_{\Omega_x} \bar{C}_p(\mathbf{i}) d\omega_{\mathbf{i}}. \quad (4.11)$$

Le calcul de luminance consiste alors en une somme d'intégrales de lobes de cosinus. Cette approche est rapide et efficace : les produits et les intégrales de lobes sont obtenus à partir de tables pré-calculées, et les seuls calculs mis en œuvre lors du calcul d'une image sont des interpolations et des sommes de scalaires.

La plupart des FDRB peut être exprimée en utilisant des sommes de lobes de cosinus [Lafortune 1997, Adrià 2009]. De plus, l'éclairage direct dû à une source se décrit facilement avec un ou quelques lobes. La seule difficulté dans notre expression du rendu réside ainsi dans le terme de visibilité, décrit dans le paragraphe suivant.

4.3.1.3 Visibilité et lobes de cosinus

Une visibilité totale sur un hémisphère s'exprime par un lobe d'exposant nul, et d'intégrale unitaire. L'occultation est alors la suppression d'une partie visible, et donc nécessite une différence. Pour un occulteur donné nous avons :

$$V(x, \mathbf{i}) = 1 - f_{occ}(x, \mathbf{i})$$

Notre méthode utilise des primitives dont l'occultation est directement transformable en lobes de cosinus : des sphères. Pour une sphère de centre y et de rayon r , la fonction f_{occ} s'exprime trivialement à l'aide d'un lobe, fonction de la direction xy et de r . L'occultation due à un objet est ainsi une différence entre deux lobes.

Une méthode naïve permettant de gérer N occulteurs consiste à multiplier les actions de chacun. Malheureusement, cela conduit très rapidement à une explosion combinatoire du nombre de lobes. Ainsi, pour N occulteurs il faut 2^N lobes :

$$V_N(\mathbf{i}) \approx \prod_{k=1}^N (1 - C_{occ}^k(\mathbf{i})) = \sum_{j=1}^{2^N} C_{(s_j, \mathbf{a}_j, e_j)}^j(\mathbf{i})$$

Il est alors nécessaire de réduire le nombre de lobes utilisés. Pour cela, il faut tenir compte de la superposition de l'occultation engendrée par des lobes indépendants : deux occulteurs bloquant les mêmes droites nécessitent en réalité un unique lobe.

Pour exprimer cette superposition, nous utilisons un masque, prenant la forme d'un *quadtree*. Le processus se déroule en deux étapes. Dans la première, chaque occulteur est projeté sur le masque, jusqu'à une profondeur maximale. La seconde étape construit une liste de lobes pour chaque feuille du *quadtree* contenant au moins un objet projeté. En pratique, nous n'utilisons que les feuilles complètement masquées par un ou plusieurs occulteurs. Un exemple de projection est proposé sur la figure 4.13.

Le résultat est donc un ensemble de lobes indépendants, permettant de maîtriser l'explosion combinatoire.

4.3.1.4 Résultat en rendu interactif

Cette méthode de rendu interactif à base de lobes de cosinus a été implantée en utilisant Cuda pour la parallélisation. Les résultats publiés dans [Meunier 2010] montrent que les approximations effectuées pour les calculs de produits ou d'intégrations de lobes sont peu perceptibles. Le rendu est donc d'une qualité visuelle convaincante.

Les performances en terme de nombre d'images par seconde dépendent des scènes. La figure 4.14 proposent deux rendus relativement simples sur une carte NVidia 285 GTX, pour lesquels la position des objets et de la caméra sont modifiables dynamiquement par l'utilisateur.

D'autres résultats soulignent néanmoins que le principal défaut de notre méthode est le

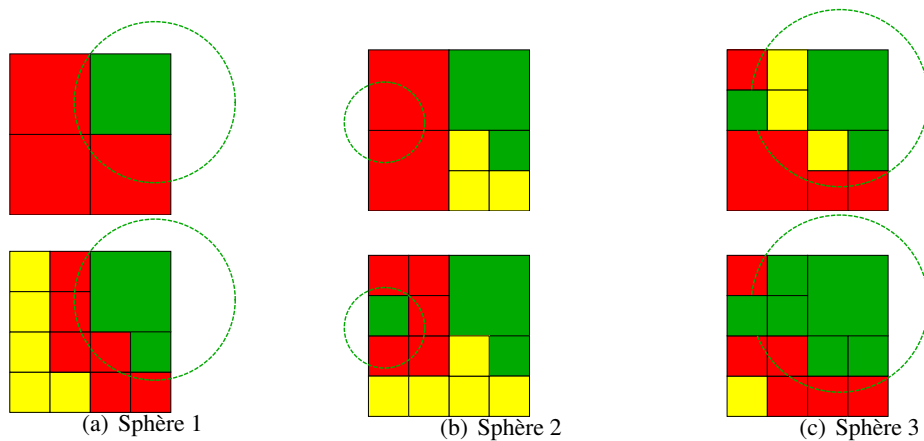


FIGURE 4.13 – Construction d'un masque de visibilité de profondeur 2 à partir de trois sphères. Les feuilles représentent : en vert, une portion non visible ; en rouge, une partie à subdiviser ; en jaune, un morceau non modifié.

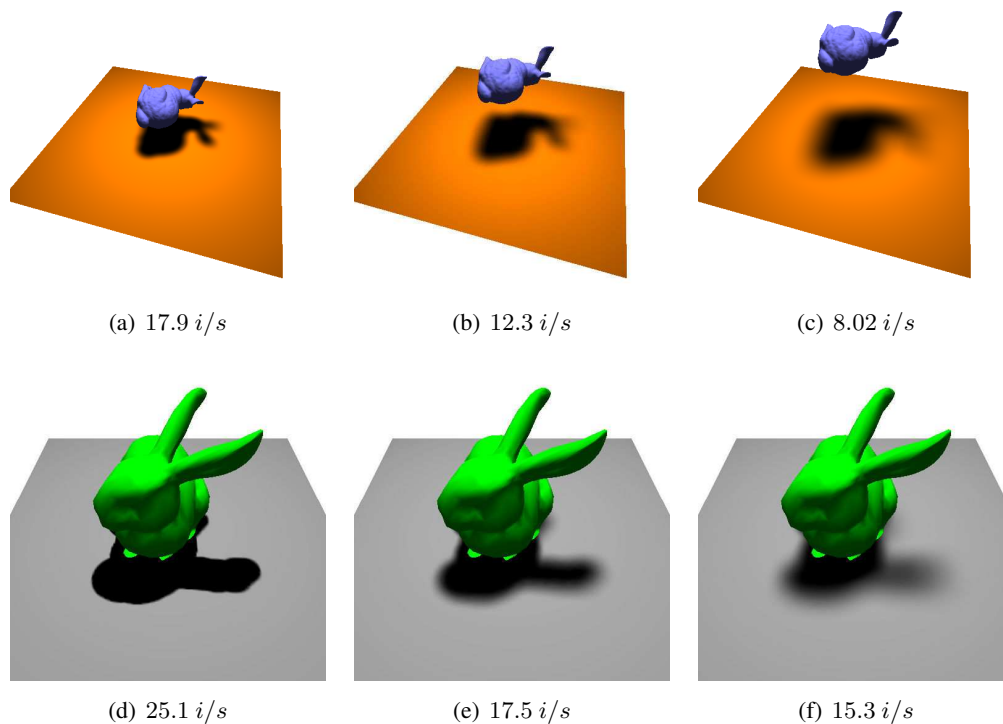


FIGURE 4.14 – Les ombres sont dures ou douces en fonction de la distance de l'objet au plan (en haut), ou de la taille de la source lumineuse (en bas). Les temps de calcul sont indicatifs.

traitement de la visibilité. Nous travaillons depuis sur un calcul d'occultation sur GPU, afin de réduire le temps du rendu y compris dans des configurations complexes.

4.3.2 Visualisation interactive de bâtiments

Ces derniers travaux concernent la première partie de la thèse de Maxime Maria, démarrée en Octobre 2012 au département SIC, en collaboration avec Sébatien Horna. La modélisation à base topologique est exploitée afin de générer automatiquement une partition spatiale qui, dans des bâtiments, permet une visualisation particulièrement rapide. Ces travaux sont en cours de soumission en revue internationale.

4.3.2.1 Introduction

En informatique graphique, la notion de rayons est introduite dès les travaux d'Appel [Appel 1968] pour calculer la visibilité point à point. Ils sont ensuite largement utilisés pour le rendu [Whitted 1980, Cook 1984, Glassner 1989]. Par échantillon de pixel, la complexité initiale est en $\mathcal{O}(N)$, avec N primitives géométriques d'une scène quelconque. Ce nombre de primitives dépasse bien souvent le million, voire le milliard.

Beaucoup de travaux réduisent cette complexité. Certains proposent des structures *accélé-ratrices*, que nous classons selon deux catégories. Ainsi existent d'une part les partitions spatiales : celles régulières [Fujimoto 1986] ou hiérarchiques [Havran 1999], les arbres BSP [Fuchs 1980] ou encore les arbres *kd* [Bentley 1975]. D'autre part, se trouvent les partitions d'objets telles que les BVH [Rubin 1980, Kay 1986] ou encore les BIH [Wächter 2006].

D'autres méthodes d'accélération regroupent les rayons en faisceaux [Heckbert 1984, Ghazanfarpour 1998], avec des cônes [Amanatides 1984] ou des pinceaux [Shinya 1987], ou encore en utilisant les instructions vectorielles des processeurs modernes [Wald 2001, Wald 2006, Reshetov 2005, Mahovsky 2006, Wald 2007]. D'autres s'intéressent à la parallélisation pour les processeurs embarqués sur cartes graphiques (GPU) [Purcell 2002, Foley 2005, Popov 2007, Horn 2007, Günther 2007, Aila 2009, Aila 2012].

Certains travaux s'intéressent cependant à des scènes particulières, telles que les bâtiments. Dans ces derniers, la géométrie est composée de pièces avec de (relativement) petites ouvertures. Elle est utilisée pour générer une partition spatiale particulière : les *cellules et portails* (cf. figure 4.15), ou CeP en abrégé. À l'intérieur d'une cellule (une pièce), il devient facile de trouver la plus proche intersection : soit c'est un bord de cette cellule, soit elle se trouve dans une cellule voisine, accessible via un portail.

La construction d'un modèle CeP reste relativement difficile, et bien souvent les méthodes n'arrivent pas à calculer les cellules optimales en fonction d'un modèle géométrique donné [Airey 1990, Teller 1991b, Meneveaux 1998]. Dans [Fradin 2005], la construction est automatiquement générée à partir d'un modèleur de bâtiments à base topologique [Lienhardt 1989]. En ajoutant une notion de sémantique à cette modélisation, il devient possible d'automatiser la génération de la structure CeP [Horna 2009].

Partant de ces derniers travaux, nous proposons de construire une structure particulièrement rapide à parcourir. Elle est capable d'inclure d'autres types de structures accélératrices, pour permettre la visualisation interactive de modèles particulièrement complexes.

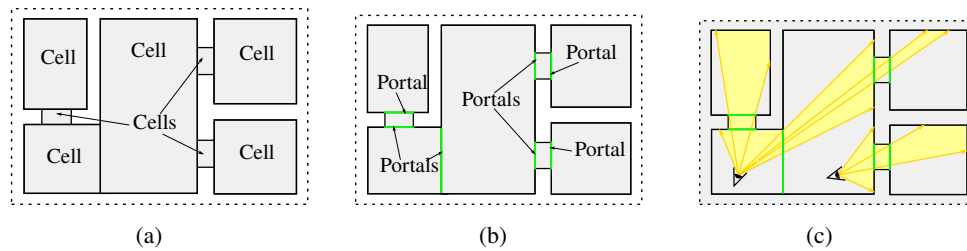


FIGURE 4.15 – Structure de type *cellule et portail* en dimension 2 : (a) les pièces et l'intérieur des portes sont des cellules ; (b) un portail appartient à deux cellules ; (c) les rayons traversent les portails pour aller dans les cellules voisines.

4.3.2.2 Partition spatiale

Notre structure possède une particularité très importante pour son exploitation : elle englobe l'ensemble des relations topologiques entre les éléments la composant. Traditionnellement, dans une structure CeP, les relations entre les volumes sont utilisées afin de passer d'une cellule à une autre. Nous utilisons en plus les relations entre faces et entre arêtes.

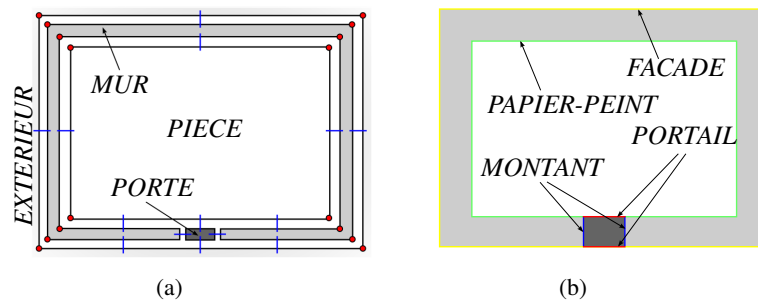


FIGURE 4.16 – Calcul de la sémantique d'une face : (a) exemple de bâtiment ; (b) une face incidente à deux volumes *PIECE* ou *PORTE* ou *EXTERIEUR* devient un *PORTAIL*, et prend sinon une autre sémantique spécifique (*PAPIER-PEINT*, *FACADE*, *MONTANT*, etc.).

La modélisation d'un bâtiment est non seulement topologique et géométrique, mais elle est aussi sémantique. Cette dernière est utilisée pour générer les portails (cf. figure 4.16). Par exemple, deux volumes adjacents ayant respectivement une sémantique *PIECE* et *PORTE* deviennent deux cellules ayant un portail en commun. Autre exemple : un mur et une pièce auront leurs faces communes étiquetées d'un côté *BRIQUE* et de l'autre *PAPIER-PEINT*.

Notre moteur de rendu interactif reçoit directement les données nécessaires pour le rendu, à savoir la liste des volumes, les sémantiques des faces, la position des objets (les meubles, etc.), ou encore la position des sources lumineuses.

Notons que tous les volumes sont convexes. Cette condition est nécessaire à notre algorithme de rendu.

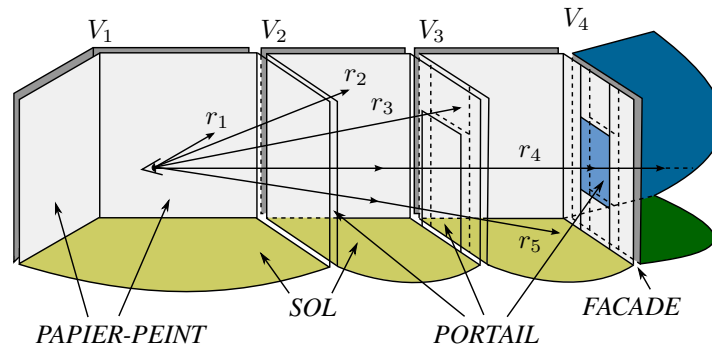


FIGURE 4.17 – Recherche de l'intersection la plus proche pour des rayons dont l'origine est dans V_1 : r_1 y trouve une intersection ; r_2 traverse un portail et s'arrête dans V_2 ; r_3 va dans V_3 après avoir traversé deux portails et V_2 ; enfin r_4 sort du bâtiment et va dans V_4 .

4.3.2.3 Recherche de la plus proche intersection

Le rendu interactif repose sur un algorithme efficace de recherche de la plus proche intersection d'un rayon. Notre structure s'utilise comme un CeP classique (*cf.* figure 4.17). Néanmoins, la connaissance de la topologie complète nous permet d'optimiser son fonctionnement. En effet, la modélisation utilise des plans architecturaux de dimension 2, qui sont ensuite extrudés verticalement pour obtenir un modèle en dimension 3. Il est ainsi possible de séparer le calcul en deux étapes, chacune en dimension 2.

L'algorithme de recherche de la face de sortie d'un rayon est illustré sur la figure 4.18. La première étape recherche la section verticale de sortie du rayon, et fonctionne horizontalement en dimension 2. La seconde étape recherche la face de sortie verticale, toujours en

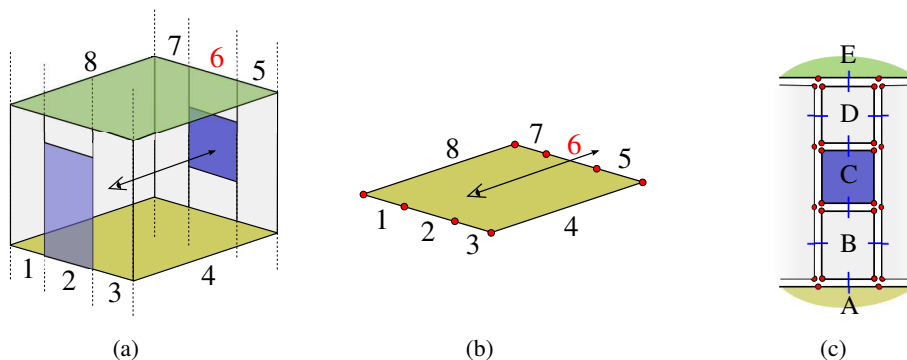


FIGURE 4.18 – Intersection d'un rayon *dans* d'un volume : (a) le résultat attendu est un portail de la section vertical 6 ; (b) première étape en dimension 2, recherche de la section verticale. (c) seconde étape en dimension 2 (verticalement), recherche de la face de sortie.

dimension 2. Ces deux étapes utilisent uniquement la notion d'orientation de droites dans \mathcal{E}_3 . En pratique, la projection en dimension 2 annule quelques-unes des coordonnées des droites, simplifiant les calculs.

Cet algorithme particulièrement simple est néanmoins très efficace. Nous l'utilisons pour les rayons primaires (issus de la caméra), et les rayons secondaires (réfléchis ou pour l'éclairage direct). Un élément-clé dans cette efficacité est la connaissance de la cellule de départ du rayon : il est inutile de descendre dans un arbre ou dans une quelconque hiérarchie pour démarrer le processus. Les rayons primaires issus de la caméra démarrent tous depuis le volume contenant la caméra. Les rayons secondaires démarrent là où le rayon précédent s'arrête. Notre structure n'étant utilisée que localement, elle en est d'autant plus efficace.

Notre modèle permet d'utiliser des objets quelconques, comme des meubles, des accessoires de bureaux, *etc.* Lors de la modélisation, les objets en question sont directement intégrés dans le modèle : les boîtes englobantes des objets deviennent des cellules du modèle. Il est donc possible de traverser via un BVH optimisé sur GPU [Aila 2012] les boîtes englobantes afin de calculer l'éventuelle intersection du rayon avec l'objet.

4.3.2.4 Résultats

Notre structure accélératrice est implantée à la fois sur CPU et sur GPU. Nous l'avons expérimentée sur les trois scènes décrites dans la table 4.2 et illustrées sur la figure 4.19. Les nombres de volumes (ou cellules) sont relativement modestes, bien qu'ils correspondent à des bâtiments relativement complexes. L'insertion d'objets dans le modèle augmente légèrement le nombre de volumes, et surtout de polygones (jusqu'à 21 millions).

Scenes	# pièces	# cellules		# polygones		# lumières
		(vide)	(objets)	(vide)	(objets)	
HOUSE	8	510	577	1166	1.005.454	8
SP2MI-3	46	1923	2510	8557	8.956.708	55
SP2MI-o	135	7243	6722	29.397	21.491.046	200

TABLE 4.2 – Caractéristiques de trois scènes de tests

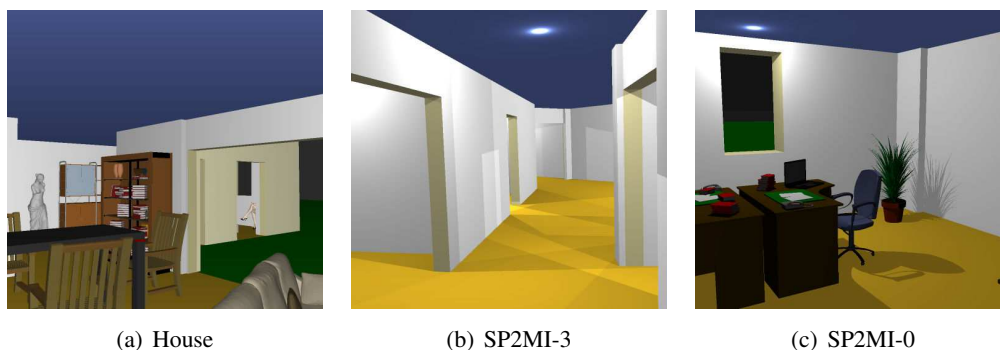


FIGURE 4.19 – Exemples de rendus pour les trois scènes de test.

Scenes		HOUSE		SP2MI-3		SP2MI-o	
Results		IPS	Mrays/s	IPS	Mrays/s	IPS	Mrays/s
CPU	Ray casting	16.98	17.8	14.03	14.71	14.87	15.6
	Packet casting	401.85	421.37	384.13	402.79	403.71	423.33
GPU	Ray casting	1353.23	1418.96	1084.48	1137.16	1118.48	1172.81
	Specular	1034.61	1293.07	723.81	988.3	717.25	1034.7
	Lighting	439.47	1438.59	156.77	988.81	163.35	1124.92
	Objects	183.95	192.89	246.19	258.15	208.04	218.15
	Objects & Lighting	64.46	163.48	27.87	180.66	28.9	167.37

TABLE 4.3 – Résultats : nombre d’images par secondes (IPS) et nombre moyen de millions de rayons traités par seconde (Mrays/s) par algorithme (CPU est un Core i7, GPU est de type NVidia GTX 680).

Nous avons sept modes de rendus, deux sur CPU et cinq sur GPU. Les performances sont données dans la table 4.3, avec un rendu de taille 1024×1024 , et des FDRB de type Lambert.

La méthode CPU est particulièrement efficace avec un rendu par faisceaux, grâce à la cohérence du parcours inhérente à un modèle CeP. Sur GPU, les performances sont supérieures. Lancer des rayons réfléchis a un faible impact, là aussi en raison de la cohérence de ceux-ci. L’ajout de sources lumineuses (de 8 à 200, cf. table 4.2) réduit inévitablement les performances ; néanmoins en pré-traitant la visibilité des sources depuis chaque volume, nous arrivons à maintenir des performances acceptables. De plus, et malgré le manque de cohérence des rayons lumineux pour différentes sources, la réutilisation des informations pré-calculées de visibilité permet de conserver un nombre de rayons lancés par seconde très correct.

Le rendu avec objets fait chuter les performances en raison de l’utilisation de BVH (surtout avec les lumières, notamment parce que nous ne réutilisons par encore les pré-calculs de visibilité). Notons qu’un outil basé sur un BVH [Aila 2012] a des performances inférieures pour la scène *HOUSE* (de l’ordre de 100 IPS), et ne peut rendre les deux autres scènes (trop de polygones).

4.4 Conclusion

Ce chapitre a rappelé nos différents travaux dans le domaine de l’informatique graphique. Une bonne partie d’entre eux est une application directe d’un calcul d’occultation. Ainsi le rendu avec ombres douces repose soit sur un arbre d’occultation, soit sur l’évaluation paresseuse de l’occultation. De même, la réduction des biais dans les cartes de photons utilise l’évaluation paresseuse de l’occultation. Le calcul d’occlusion ambiante utilise aussi une évaluation paresseuse, mais d’un arbre de visibilité. Dans chaque cas, le calcul d’occultation ou de visibilité permet soit d’accélérer les traitements en raison de la réutilisation des calculs, soit d’améliorer la qualité du rendu de façon notable, soit les deux.

La dernière partie de ce chapitre a proposé deux autres applications qui, si elles ne reposent pas sur nos calculs d’occultation ou de visibilité, traitent malgré tout de la visibilité. Avec

les lobes de cosinus, nous exprimons l'ETL afin d'effectuer un rendu dynamique avec éclairage direct. Notre nouvelle partition spatiale issue de la modélisation à base topologique permet de calculer très rapidement la primitive la plus proche traversée par un rayon donné, permettant des temps de rendu particulièrement faibles.

Tous ces travaux furent réalisés dans le cadre des thèses de doctorat de F. Mora (2006), S. Charneau (2007), R. Perrot (2012), et M. Maria (soutenance prévue en 2015). Ils ont été publiés dans 4 revues internationales, 2 autres articles étant en cours de soumission.

Travaux en Simulation de Propagation d'Ondes

Sommaire

5.1 Introduction à la simulation radio	95
5.1.1 Problématiques d'une simulation radio	95
5.1.2 La visibilité pour optimiser une simulation	99
5.2 Simulations en dimension 2	100
5.2.1 Graphe de visibilité en dimension 2	101
5.2.2 Exploitation du graphe de visibilité	103
5.3 Simulations en dimension 3	105
5.3.1 Nos deux simulateurs	105
5.3.2 Radars à travers les murs	110
5.3.3 La simulation en tunnels	115
5.4 La simulation pour l'optique sans fil	118
5.4.1 Nouvelle équation de transfert lumineux	119
5.4.2 Méthodes de Monte-Carlo	119
5.5 Conclusion	122

Dans ce chapitre, nous présentons une vue d'ensemble des problématiques associées aux simulations radioélectriques, puis nos travaux sur la propagation radio en dimension 2, la simulation radio en dimension 3, et enfin la simulation optique.

5.1 Introduction à la simulation radio

Cette section présente les spécificités d'une simulation de propagation d'ondes radio.

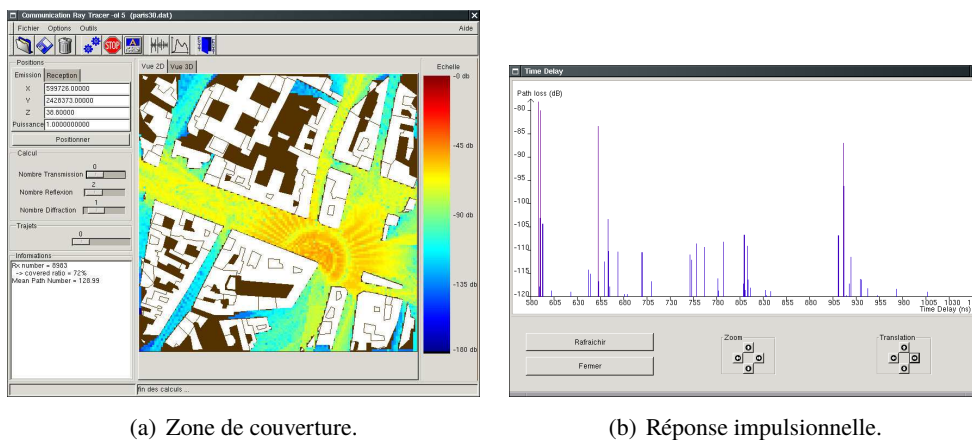
5.1.1 Problématiques d'une simulation radio

Mon implication dans la simulation de propagation d'une onde radio remonte à mon doctorat. Lors de mon recrutement comme maître de conférences à l'Université de Poitiers, l'objectif était entre autre de poursuivre des recherches sur cette thématique, afin de renforcer la collaboration entre les équipes "informatique graphique" et "simulation radio" du laboratoire SIC. Le développement d'un premier simulateur radio était engagé dès la fin de ma thèse ; il s'est naturellement poursuivi après mon recrutement.

5.1.1.1 Motivations

Nous distinguons deux enjeux principaux d'une propagation d'ondes radio : la planification et le déploiement d'un ensemble de stations fixes, et la caractérisation du canal.

La planification consiste à prédire l'atténuation d'un signal en tous points d'une scène donnée (une commune, une ville, un bâtiment). L'objectif est de permettre le placement d'un ensemble d'émetteurs dans cette scène, en minimisant les conflits de réception entre émetteurs, en utilisant un minimum d'émetteurs, en respectant les normes de puissance d'émission maximale, et bien sûr en garantissant une qualité minimale de service en tous points de la zone de déploiement.



(a) Zone de couverture.

(b) Réponse impulsionnelle.

FIGURE 5.1 – Deux résultats complémentaires de simulations : (a) en bande étroite, seule la puissance compte ; (b) en bande large, la réponse du canal dans le temps est obtenue.

Dans ce contexte, l'atténuation du signal est un calcul en *bande étroite* (cf. figure 5.1(a)) : le résultat est généralement une puissance totale reçue divisée par la puissance totale émise, exprimée en décibel. Le calcul sur une zone d'étude donnée doit être rapide, car le placement des émetteurs est effectué par des algorithmes itératifs, convergeant lentement vers une solution acceptable.

La seconde utilisation d'une simulation de propagation d'ondes radio est la caractérisation du canal. Un exemple d'application est la recherche d'algorithmes de communication efficaces dans des environnements réalistes. Un tel algorithme permet le dialogue entre des terminaux connectés à un canal de communication commun. Le codage de l'information concerne ici les couches basses du modèle OSI, aussi bien physique que trame. Une caractérisation du canal réaliste fournit les informations nécessaires pour simuler le comportement des algorithmes, afin de permettre leurs évaluations.

Prenons un exemple volontairement simple afin d'illustrer notre propos, avec un modèle de communication de type *On/Off*. Chaque bit de l'information à transmettre est codé sur un intervalle de temps fixe δT , en émettant par exemple 5 mW pour un bit à 1 et 0 mW pour un bit 0, le tout à une fréquence d'émission de $1/\delta T$. L'objectif de la simulation du canal est de vérifier la robustesse du codage, *i.e.* valider que les bits émis sont correctement reçus, sans

erreur. Avec notre protocole simpliste, cela concerne *a priori* la puissance reçue tous les δT . Cependant, les choses sont plus complexes : un bit étant émis sur l'intervalle de temps δT , sa réception est attendue sur le même intervalle (sinon des bits seraient perdus). Or, les mécanismes de propagation sont tels que le signal risque d'arriver sur un temps beaucoup plus long que celui attendu, engendrant des interférences inter-symboles.

La caractérisation du canal consiste donc, pour ce genre de contexte, à simuler la propagation d'une impulsion de type Dirac, et à étudier sa réception étalée dans le temps : la *réponse impulsionnelle* du canal (cf. figure 5.1(b)). Nous la représenterons soit comme l'ensemble des trajets reçus (*i.e.* une puissance et un temps de propagation), soit sous une forme simplifiée avec une puissance par intervalle de temps (*i.e.* un histogramme). Avec notre modèle de communication On/Off, si la réponse impulsionnelle fait apparaître une réception mesurable (au delà du bruit de fond) à un instant T_0 , puis une seconde réception mesurable à un instant $T_0 + 2\delta T/3$, alors la simulation de l'émission des bits 1010 engendrera la réception erronée du train 1111.

Une chaîne complète de simulation typique inclut des algorithmes de correction d'erreur, afin de quantifier le nombre d'erreurs en réception, discriminant celles qui sont corrigées, celles simplement détectées mais sans correction, et celles non détectées par la chaîne de transmission.

5.1.1.2 Les besoins en radio

Les communications radioélectriques reposent sur des fréquences relativement faibles en comparaison de l'optique (quelques dizaines de KHz pour la radio, à 60 GHz pour le Wifi, norme IEEE 802.11 ad). Ainsi, les phénomènes électromagnétiques prépondérants sont-ils différents. Une simulation radio considère les mécanismes de réflexion spéculaire, de réfraction, de diffraction. L'information n'est pas représentée par une puissance, mais par un champ électromagnétique, autorisant la simulation des interférences et de la polarisation.

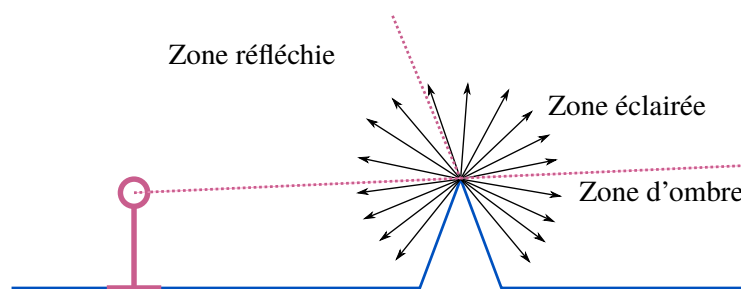


FIGURE 5.2 – Diffraction par un dièdre dans un modèle asymptotique de propagation.

Notre simulation de propagation d'ondes radio est dite *asymptotique*, car elle repose sur une théorie simplifiant les équations de Maxwell. Elle se base physiquement sur l'Optique Géométrique classique (OG), et la Théorie Uniforme de la Diffraction (TUD) [Kouyoumjian 1974, Balanis 1989]. Si la première est classique, la seconde mérite quelques

explications. La TUD est construite sur la Théorie Géométrique de la Diffraction (TGD) [Keller 1962], et en améliore les aspects physiques. La TGD prolonge l'OG en prenant en compte le phénomène de diffraction. Ainsi, une des faiblesses de l'OG qui est de prévoir un champ électromagnétique nul dans l'ombre géométrique des objets est supprimée. Pour cela de nouveaux rayons viennent s'ajouter aux rayons de l'OG : les rayons diffractés (cf. figure 5.2).

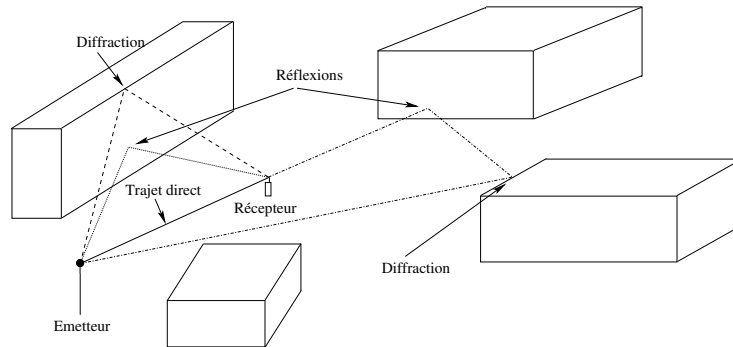


FIGURE 5.3 – Quelques chemins dans une propagation par trajets multiples.

Ces nouveaux rayons sont émis lorsqu'une onde (et donc un rayon incident) arrive sur une discontinuité de surface, tel qu'un dièdre. L'arête du dièdre se comporte alors comme une source secondaire, qui émet dans un ensemble précis de directions. Les rayons diffractés satisfont le principe de Fermat généralisé, qui stipule que la durée du parcours d'une onde est stationnaire. Ainsi, pour un unique rayon incident, le sommet d'un dièdre émet une infinité de nouveaux rayons diffractés localisés sur un cône dont l'angle d'ouverture correspond à l'angle d'incidence. Un exemple de propagation impliquant des trajets simplement réfléchis et simplement diffractés est proposé sur la figure 5.3.

Une simulation de propagation doit ainsi respecter les contraintes suivantes :

- gérer la diffraction sur un ensemble fini de sources secondaires définies géométriquement comme des discontinuités de surfaces ;
- permettre la réflexion purement spéculaire, engendrant un nombre fini de chemins de propagation entre un émetteur et un récepteur donné ;
- reposer sur un calcul physique vectoriel, soit en bande large pour produire la réponse impulsionnelle du canal, soit en bande étroite et fournissant une simple puissance reçue.

Ces contraintes font que la simulation radio diffère fondamentalement des méthodes de rendu en synthèse d'images. Notamment, une simulation radio vise à calculer l'ensemble des trajets existants pour certaines configurations d'interactions physiques (e.g. pour n réflexions, ou pour 1 réflexion puis 1 diffraction, pour l'inverse, etc.), et surtout sur une durée de propagation maximale donnée. Néanmoins, la simulation radio et le rendu graphique ont aussi deux points communs :

- la propagation est effectuée le long de rayons ; le noyau géométrique d'un simulateur radio est donc fortement similaire à celui d'un moteur de rendu, notamment pour tous les tests d'intersection.

– la visibilité, et principalement l’occultation, joue un rôle dans les deux cas.

5.1.2 La visibilité pour optimiser une simulation

5.1.2.1 Calculer les chemins

La simulation radio repose sur le calcul déterministe de tous les chemins de propagation possibles, pour des combinaisons d’interactions données. Cet aspect est lié à la nature des interactions physiques considérées, notamment la réflexion purement spéculaire, et la diffraction selon un cône. Ce calcul déterministe possède une complexité particulièrement forte, rendant tout algorithme naïf impraticable.

Pour exprimer cette complexité, nous partons de l’exemple suivant : soit une combinaison de type $RDRR$ concernant tous les chemins partant de l’émetteur, subissant une réflexion sur une face, puis une diffraction par une arête, puis deux réflexions consécutives sur deux faces différentes, et finalement arrivant au récepteur. Considérant une scène composée de n_f faces planes et de n_a arêtes, alors une telle combinaison représente $n_f \times n_a \times n_f \times (n_f - 1)$ chemins possibles, soit approximativement $n_f^3 n_a$ possibilités. Par extension et en ignorant l’ordre des interactions (*i.e.* combinaison de d parmi $r + d$), une combinaison contenant r réflexions et d diffractions a de l’ordre de $\mathcal{O}(n_f^r n_a^d)$ chemins potentiels. Prenons un exemple avec une simulation relativement simple dans un environnement composé de $n_f = 1000$ faces et $n_a = 1500$ arêtes. Le nombre de chemins à calculer naïvement pour 6 réflexions et 2 diffractions est alors environ $10^{3 \times 6} \times (1.5 \times 10^3)^2$, soit de l’ordre de $\mathcal{O}(10^{24})$.

Notons que le nombre d’arêtes est lié au nombre de faces ; ainsi, la complexité d’une simulation radio se simplifie en $\mathcal{O}(n^i)$, en notant $i = r + d$ le nombre total d’interactions.

Pour un ensemble de faces et d’arêtes donné, calculer un chemin est un processus complexe. Il commence par déterminer les positions des points d’interaction sur chaque élément géométrique, *i.e.* les points de réflexion et de diffraction. Cette étape est effectuée en respectant le principe de Fermat étendu, par une minimisation de la longueur du chemin en question. A ce moment, la réflexion est sur le plan support d’une face, la diffraction sur la droite support d’une arête. Chaque chemin passe ensuite par une phase de validation, consistant à vérifier que les points d’interaction sont bien sur les faces et arêtes concernées. La dernière étape de ce processus est enfin la phase d’occultation, consistant à vérifier que les points d’interaction consécutifs d’un chemin sont en visibilité deux à deux.

La complexité d’un tel calcul naïf est clairement problématique. Bien entendu, une méthode de calcul efficace d’un chemin donné est essentielle, mais elle n’influe pas sur la complexité globale de l’algorithme qui reste en $\mathcal{O}(n^i)$. La recherche d’une méthode praticable nous a conduit très tôt à considérer la visibilité comme une solution.

5.1.2.2 Le cas de la visibilité

La visibilité joue un rôle important dans un simulateur de propagation d’ondes radio. Fondamentalement, il ne s’agit pas de connaître la plus proche intersection, mais de résoudre un

problème d'occultation : partant d'un chemin potentiel formé d'une suite de points sur des faces (réflexion ou transmission) ou des arêtes (diffraction), sa validation consiste à vérifier la non occultation de chacun de ses couples de points successifs. Dans un moteur de rendu, un problème similaire se retrouve notamment lors du calcul d'illumination direct.

Ce calcul est effectué en utilisant une structure accélératrice classique, issue de la synthèse d'images. Notre premier simulateur (CRT) utilisait au choix une grille régulière [Fujimoto 1986], un octree [Glassner 1984] ou un kdtree [Bentley 1975]. Le simulateur actuel (RaPSor) utilise une structure de type *Bounding Interval Hierarchy* [Wächter 2006]. Les calculs d'occultation en versions pré-calcul et paresseuse introduits dans le chapitre 3 ne sont pas pertinents ici, car ils nécessitent une certaine cohérence des requêtes pour être efficace. Or, cette condition n'est pas respectée par un moteur de simulation radio. Notre utilisation de la visibilité pour réduire la complexité d'une simulation est ailleurs : nous nous intéressons à la réduction du nombre de chemins calculés et non pas à l'accélération du calcul d'un unique chemin.

Le paragraphe précédent indique que la complexité d'une simulation rend très rapidement une solution naïve impraticable. L'idée principale développée dans nos travaux sur la simulation est que la visibilité permet d'obtenir des solutions praticables. Il est relativement simple d'organiser les calculs sur les chemins potentiels sous la forme d'un arbre. La racine est l'émetteur. Le premier niveau contient $n_f + n_a$ fils. Le second niveau répète le premier, à savoir que chaque nœud de niveau 1 possède $n_f + n_a$ fils¹. Et ainsi de suite pour tous les niveaux inférieurs. Un algorithme de simulation naïf consiste alors à parcourir un tel arbre jusqu'à une profondeur donnée, et à connecter et traiter chaque nœud rencontré avec le récepteur (y compris la racine pour obtenir le trajet direct).

Si elle ne modifie en rien la complexité, cette vision arborescente souligne la répétition inhérente à la simulation : ainsi, peu importe la longueur d'un chemin, une même succession de deux interactions (il y en a quatre : RR, RD, DR, DD) revient toujours aux mêmes couples de faces ou d'arêtes impliquées. L'idée d'une optimisation basée sur la visibilité est alors immédiate : si un calcul prouve que deux éléments d'un couple ne peuvent se voir, alors nous pouvons supprimer définitivement l'arête les reliant.

Bien entendu cette idée n'est ni nouvelle ni propre à la simulation d'ondes, se retrouvant aussi en informatique graphique [Bouatouch 1992]. C'est néanmoins celle que nous avons suivie, et qui est présentée au fil de ce chapitre.

5.2 Simulations en dimension 2

Dans le cadre du contrat de recherche *Topase* avec France Télécom Recherche & Développement, nous nous sommes intéressés à une simulation de propagation radio en dimension 2 [Aveneau 2002, Combeau 2002a]. L'objectif est la prise en compte d'environnements

1. Normalement ce devrait être $n_f + n_a - 1$, pour éviter la répétition absurde consistant à soit réfléchir deux fois de suite sur la même face, soit diffracter deux fois de suite sur la même arête ; notre propos étant la complexité de la méthode, nous ignorons sciemment ce truisme.

urbains de grandes tailles. Pour éviter l'explosion combinatoire des calculs, ceux-ci sont réduits dans deux plans orthogonaux, l'un horizontal et l'autre vertical. Le premier permet de grouper les calculs pour l'ensemble de récepteurs coplanaires. Le second permet d'affiner localement le calcul pour un récepteur donné.

Ces travaux furent menés pendant la thèse de Pierre Combeau, soutenue en 2004, en collaboration avec Rodolphe Vauzelle du laboratoire SIC.

5.2.1 Graphe de visibilité en dimension 2

Dans [Agelet 2000], Agelet calcule un graphe de visibilité en dimension 2, reposant sur des calculs numériques relativement coûteux. Dans [Athanasiadou 2000], Athanasiadou présente un calcul similaire mais simplifié, générant des graphes trop gros. L'utilisation de ces derniers nécessite alors des validations de l'occultation.

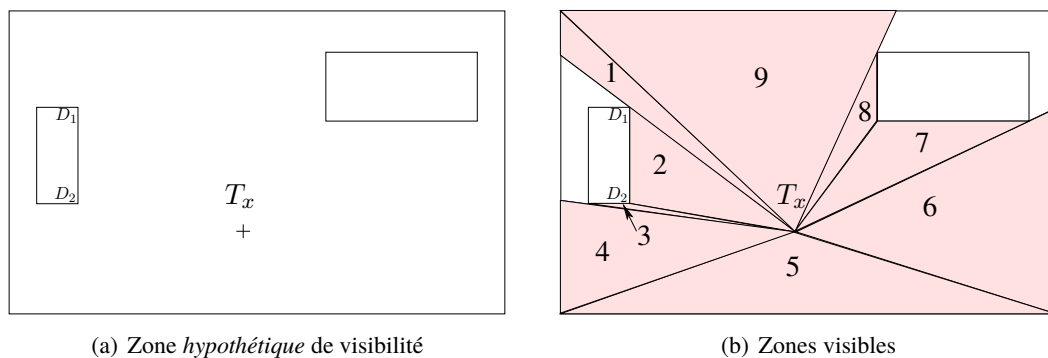


FIGURE 5.4 – Premier niveau du graphe de visibilité.

Nous proposons de calculer un graphe de visibilité reprenant les avantages de ces précédents travaux. Partant d'un émetteur et d'une fenêtre de calcul donnés, nous commençons par construire une zone hypothétique de visibilité (*cf.* figure 5.4(a)). Considérant la géométrie (les bâtiments) contenue dans cette zone hypothétique, nous la découpons, aboutissant à un ensemble de zones de visibilité correctes (*cf.* figure 5.4(b)). Le résultat est stocké dans un arbre de racine T_x , et ayant 9 fils étiquetés par chacune des zones.

Cette étape est appelée réduction. Dans une version naïve, elle peut être relativement coûteuse en terme de temps de calcul. Nous utilisons une version optimisée reposant sur la géométrie discrète [Reveillès 1995]. Une zone hypothétique est toujours définie sous forme d'un polygone convexe. Nous discrétisons le bord de cette zone en utilisant la notion de supercouverture de droite [Andres 1997]. Un parcours dans une direction voisine de celle de l'onde associée à la zone permet de trouver rapidement le découpage correct de la zone hypothétique initiale. De plus, cela permet de limiter le nombre de découpages inutiles, et donc de tendre vers un ensemble minimal de zones, synonyme de temps de calcul global minimum.

Le calcul du graphe de visibilité se poursuit sur chacune des zones obtenues au premier

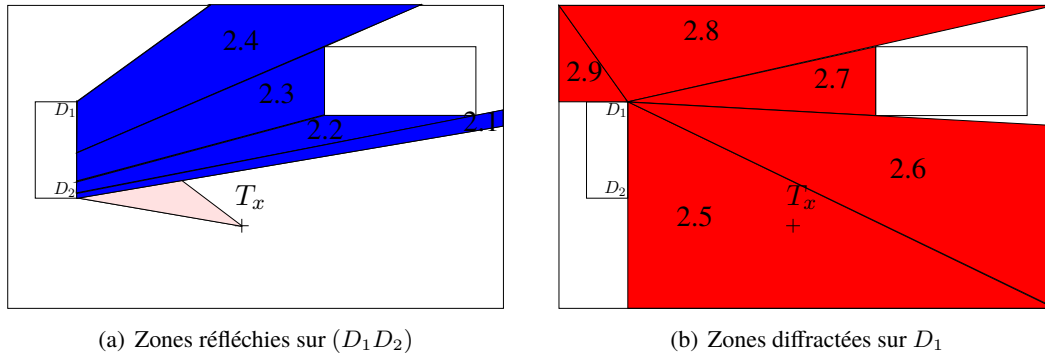


FIGURE 5.5 – Exemple de zones réfléchies ou diffractées sur la géométrie, après réduction.

niveau, de façon récursive. La figure 5.5(a) illustre le cas de zones réfléchies, la figure 5.5(b) celui de zones diffractées.

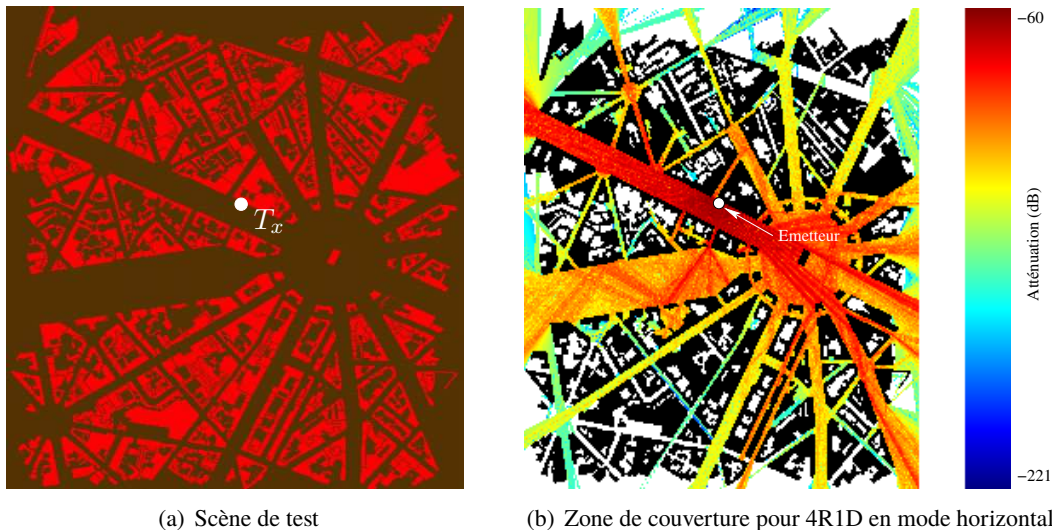


FIGURE 5.6 – Scène et zone de couverture sur la place de l'Etoile, Paris.

Ce mécanisme permet de construire un graphe de visibilité associé à la propagation d'une onde émise depuis un émetteur donné. Lorsque l'émetteur est déplacé, le calcul doit cependant être relancé. Un exemple de résultat sur la scène illustré dans la figure 5.6(a) est proposé dans le tableau 5.1 ci-dessous. L'utilisation de la géométrie discrète pour la réduction se révèle donc particulièrement efficace, à la fois en terme de temps de calcul mais aussi parce qu'elle diminue le nombre de zones réduites obtenues. Ce dernier effet est dû à l'ordre de recherche des limites d'une zone hypothétique.

Ce premier graphe de visibilité est implémenté sous forme d'un arbre. Le nombre de zones obtenues pouvant devenir très important, cette structure pose des problèmes en terme de gestion mémoire. Partant de l'observation que pour un émetteur donné (y compris les sommets diffractants) un arbre est toujours le même, nous utilisons un vrai graphe de visibilité.

Inter	Méthode naïve		Méthode discrète		Ratio	
	Temps (s)	# zones	Temps (s)	# zones	Temps	# zones
0R 0D	25''91	1990	0''10	67	259,1	29,70
1R 1D	5943''10	382483	14''28	19750	416,18	19,36
2R 1D	19351''40	1232485	37''39	60701	517,56	20,30
3R 1D	55626''33	3581384	77''82	142061	714,81	25,21
4R 1D	NC	NC	142''25	283027	-	-

TABLE 5.1 – Comparaison des méthodes de réduction naïve et discrète.

La réutilisation des calculs et de la mémoire améliore significativement les performances de l'algorithme, comme illustré dans la table 5.2. Nous considérons cette dernière méthode comme obligatoire dès qu'une diffraction est considérée. Notons enfin que la diffraction

Inter	Arbre		Graphe		Ratio	
	Temps (s)	# zones	Temps (s)	# zones	Temps	# zones
0R 0D	0''10	67	0''10	67	1	1
1R 1D	14''28	19750	8''91	14858	1,60	1,33
2R 1D	37''39	60701	18''64	37389	2,01	1,62
3R 1D	77''82	142061	34''70	75591	2,24	1,88
4R 1D	142''25	283027	58''73	134898	2,42	2,10
1R 2D	914''09	1217607	76''14	94928	12,00	12,83
2R 2D	3002''36	4541325	171''22	222930	17,54	20,37
3R 2D	NC	NC	346''37	420596	-	-
4R 2D	NC	NC	667''26	701164	-	-

TABLE 5.2 – Comparaison des méthodes discrète et optimisée.

depuis un sommet de la géométrie engendre un arbre indépendant de la position d'émission. Cela permet de traiter ce problème en pré-calcul, accélérant encore les performances de la méthode, ou encore de réutiliser un graphe donné lorsque l'émetteur est déplacé par l'utilisateur ou par un algorithme de placement d'émetteur.

5.2.2 Exploitation du graphe de visibilité

Le calcul d'une zone de couverture est traditionnellement réalisé à partir d'une grille de récepteurs. En chaque position de réception, un calcul de puissance reçue (bande étroite) est effectué. Les résultats sont ensuite sauvegardés, et affichés sous différents formats tels qu'une image (cf. figure 5.6(b)).

Cette méthode est directement utilisable avec le graphe de visibilité. Néanmoins, nous avons proposé une alternative, consistant à partitionner la zone de travail en cellules dans lesquelles une même série de chemins existe. Les chemins en question sont identifiés par leur combinaison d'interactions ; ils correspondent à une même branche de l'arbre couvrant du graphe de visibilité.

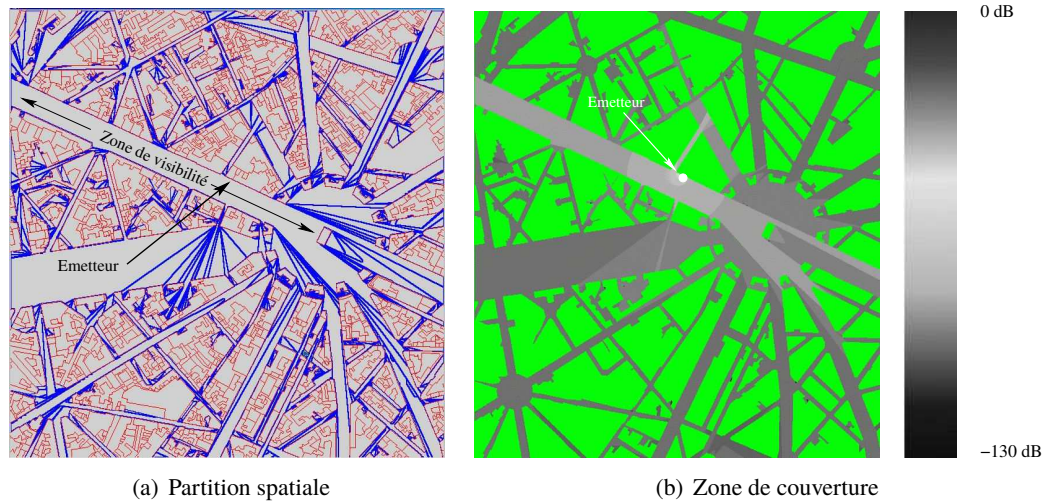


FIGURE 5.7 – Extrapolation d’une zone de couverture, place de l’Étoile, Paris.

Un tel partitionnement est illustré sur la figure 5.7. Pour chaque zone, un nombre restreint de calculs de propagation est effectué (réduction de l’ordre de 87 à 97 %), permettant d’obtenir très rapidement une carte de couverture. Notons que les calculs de propagation effectués ici sont en dimension 3, en utilisant notre moteur de simulation *CRT* [Aveneau 2000, Escarieu 2001, Aveneau 2003a, Aveneau 2004a].

Nous avons aussi utilisé directement le graphe de visibilité afin d’effectuer les calculs de propagation, en chaque point de réception de la zone de couverture. Cette méthode fonctionne en dimension 2 dans un plan horizontal ou vertical. La conjonction des deux calculs permet un modèle dit “2d et demi”. Cette méthode est optimisée en utilisant encore la géométrie discrète : dans une phase de pré-traitement, l’ensemble des zones réduites est projeté sur une grille régulière. La position de chaque récepteur correspond alors à un pixel donné de cette grille. Pour chaque récepteur, une estimation de la puissance maximale reçue doit être effectuée, en calculant les puissances associées aux trajets principaux de chaque zone traversant son pixel. Statistiquement, nous savons que la puissance finale peut être calculée uniquement en prenant en compte les trajets importants, dont l’atténuation possède un écart de moins de 18 dB avec celle du trajet principal. Nous en déduisons ainsi, pour chaque pixel, une liste ordonnée de trajets potentiels.

Pour chaque point de réception contenu dans un pixel donné, cette liste permet de limiter le nombre de zones réduites (et donc de nœuds du graphe, et donc de chemins de propagation) utilisées dans le calcul d’une puissance reçue. Cette méthode permet globalement de réduire les temps de calcul d’un facteur 3 [Combeau 2006]. Cette technique de simulation en dimension 2 a ensuite été utilisée pour la simulation du canal de propagation en lien avec la transmission d’images [Combeau 2007].

5.3 Simulations en dimension 3

Si la simulation en dimension 2 est efficace pour de grands environnements urbains, à l'intérieur des bâtiments² elle ne convient plus. Nous présentons dans cette section nos travaux en dimension 3, conduits en collaboration avec R. Vauzelle et P. Combeau du SIC.

5.3.1 Nos deux simulateurs

Au fil des ans, nous avons réalisé deux simulateurs : si le premier est la suite de mes travaux de thèse, le second en est une refonte complète.

5.3.1.1 Premier simulateur en dimension 3

Notre premier simulateur, appelé *CRT*, partage son noyau avec un moteur de rendu développé durant ma thèse. Très rapidement, les besoins différents des nombreux utilisateurs nous conduisirent à implanter un mécanisme de greffons. Ceci permit notamment à deux doctorants d'implanter une partie de leurs travaux respectifs :

- S. Reynaud, sur l'hybridation de deux méthodes, l'une basée rayon et l'autre sur les différences finies. Les simulations résultants de la seconde sont stockées, puis réutilisées dans le moteur de CRT pour simuler des objets rayonnants [Reynaud 2005a, Reynaud 2005b].
- Y. Cocheril, sur les simulations à haute fréquence (60 GHz), incluant des surfaces rugueuses. La simulation repose sur des méthodes de Monte-Carlo (shooting et bi-directionnel), et inclut des FDRB spécifiques dont les paramètres reposent sur de l'analyse d'images [Cocheril 2005a, Cocheril 2005b, Cocheril 2006].

Ces travaux n'étant pas lié à la visibilité, et n'étant pas encadrant des deux doctorants, il ne seront pas plus détaillés.

Nous décrivons ici deux greffons particuliers de CRT visant à réduire les temps de calcul d'une simulation de propagation, et en lien avec la visibilité.

Optimisation Géométrique La complexité d'une version naïve est en $\mathcal{O}(n_f^i)$, n_f étant le nombre de faces de la scène d'étude, i le nombre maximum de réflexions et de diffractions. Le §5.1.2.2 propose de modéliser les différents chemins potentiels dans un arbre. Effectuer une simulation revient alors à parcourir cet arbre virtuel. Notre optimisation géométrique consiste à élaguer une partie des fils sur un principe géométrique particulièrement simple à mettre en œuvre : nous calculons pour chaque face ou arête le faisceau englobant tous les potentiels rayons réfléchis ou diffractés, et supprimons du parcours tous les fils n'étant pas au moins en partie dans ce faisceau.

Ce calcul est effectué de la façon suivante :

- Pour les faces, le mécanisme de réflexion spéculaire respecte le principe de Fermat ; la loi de Snell-Descartes qui en découle indique que les rayons issus d'un émetteur donné de

2. L'intérieur des bâtiments est appelé *indoor*, par opposition aux environnements extérieurs dits *outdoor*.

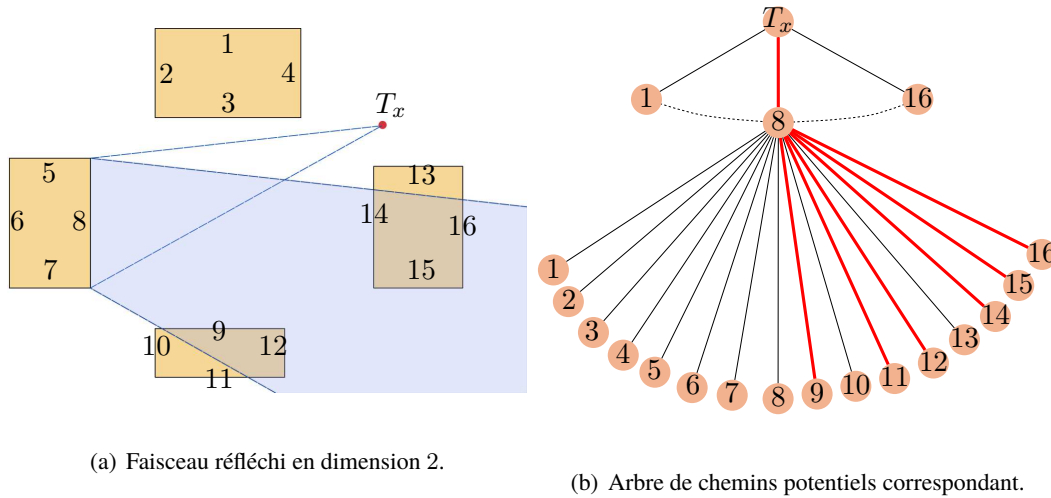


FIGURE 5.8 – Illustration en dimension 2 de l'optimisation géométrique pour des faces.

position T_x , puis réfléchis sur une face F plane et convexe de sommets f_i , sont contenus dans un faisceau convexe défini par les droites $T'f_i$, T' étant le symétrique de T par rapport à la face F . Une illustration de ce principe en dimension 2 est proposé dans la figure 5.8(a). L'impact sur l'arbre des chemins potentiels est illustré par les branches en rouge sur la figure 5.8(b). En pratique, plusieurs réflexions successives permettent d'affiner ce faisceau, réduisant d'autant le nombre de chemins potentiels à valider.

- Pour les arêtes, nous savons avec le principe de Fermat généralisé que la durée d'un trajet optique est stationnaire. La conséquence est que pour un rayon incident sur une arête, les rayons diffractés sont localisés sur un cône. Pour les rayons incidents et issus d'un même émetteur T_x , et une arête rectiligne d'extrémités les points A et B , alors l'ensemble des rayons potentiels sont sur et entre les cônes de sommets A et B , et d'angles respectifs $\widehat{AB}, \widehat{AT_x}$ et $\widehat{BA}, \widehat{BT_x}$. Nous utilisons ce principe pour vérifier que les fils du nœud contenant l'arête AB sont bien entre ces deux cônes (cf. figure 5.9). Là encore, ce principe relativement simple permet d'éliminer un grand nombre de chemins potentiels. Ces deux principes géométriques relativement simples réduisent les temps de calculs de moitié [Aveneau 2000]. Au delà d'éliminer un certain nombre de fils d'un noeud donné, ils servent surtout à éliminer des sous-arbres entiers lors du parcours de l'arbre de chemins potentiels.

Cette version purement géométrique n'inclut pas la visibilité au sens défini dans les chapitres 2 et 3, sinon l'arbre de la figure 5.8(b) aurait moins de branches en rouge (puisque 8 ne voit pas 11, 12, et 16). Néanmoins, c'est ici que réside notre intérêt original pour ces travaux, notamment dans le cadre du contrat de recherche France Télécom R&D ayant financé la thèse de F. Mora. Inclure la visibilité entre chaque couple de faces est une extension relativement simple, et reste prévue dans le futur. Par contre, possédant un faible intérêt en terme de recherche, elle a une faible priorité.

Notons que cette méthode est sans perte : les chemins potentiels supprimés ne peuvent avoir d'existence réelle, et aucun autre chemin n'est éliminé, assurant de ne rater aucun trajet.

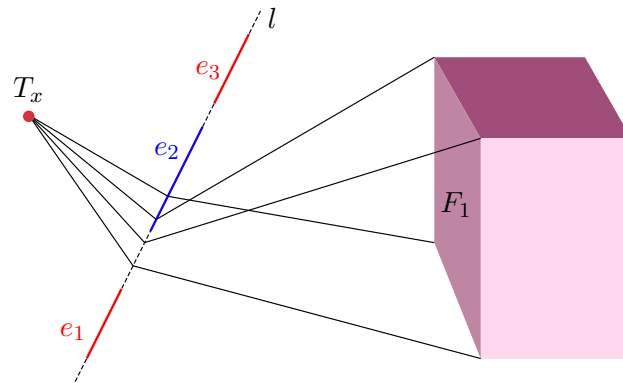


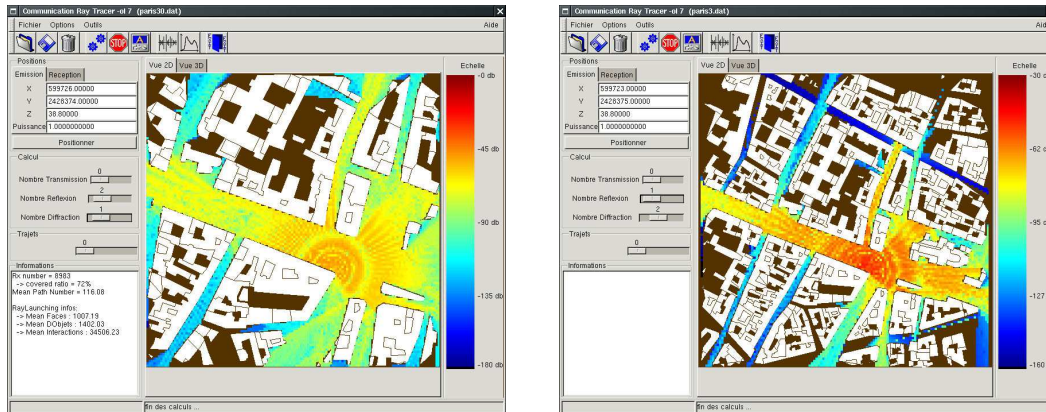
FIGURE 5.9 – Principe de l’optimisation géométrique pour des arêtes : trois arêtes sont sur la même droite l ; la diffraction sur l’arête e_2 engendre des rayons allant frapper la face F_1 , ce qui se vérifie en testant l’intersection de e_2 avec l’enveloppe convexe des rayons diffractés traversant F_1 ; par contre, aucun rayon diffracté par les arêtes e_1 et e_3 ne peut traverser la face F_1 , ce qui permet là encore de simplifier l’arbre de chemins potentiels.

Méthode oracle Cette seconde optimisation repose sur un pré-calcul *agressif* de la visibilité, au sens où certains calculs peuvent conclure à tort à une invisibilité. Le principe est le suivant : partant d’un émetteur donné, sont marquées les faces visibles soit directement soit après une ou plusieurs interactions (*i.e.* réflexion ou diffraction). Les arêtes des faces ainsi traitées sont elles aussi étiquetées. Lors du parcours du graphe, les faces et les arêtes qui ne sont pas marquées sont alors ignorées.

L’étape de pré-calcul s’appuie sur une méthode de Monte-Carlo, décrite dans [Aveneau 2004a]. Des rayons sont aléatoirement tirés depuis l’émetteur. Une recherche de la face la plus proche est effectuée ; le cas échéant, la première face traversée est marquée comme visible, tout comme ses arêtes. Ensuite et de façon récursive, le rayon est réfléchi de façon spéculaire, et depuis les arêtes de la face de nouveau rayons sont lancés en respectant le principe de Fermat généralisé (autrement dit, ils sont lancés dans le faisceau diffracté). Ce processus se poursuit ainsi récursivement, jusqu’à une profondeur maximale choisie par l’utilisateur (nombre de réflexions et de diffractions maximums). Malgré son côté très agressif, cette méthode produit d’assez bon résultats, au moins sur les premiers niveaux du graphe.

Cette méthode est une extension directe de la précédente, en ajoutant la visibilité. Néanmoins son principal inconvénient réside dans son aspect agressif : le marquage des faces n’est pas réalisé avec une méthode de visibilité exacte. Plus on descend dans le graphe, et plus il y a de risques de rater des faces ou des arêtes qui, à tort, n’ont pas été marquées comme visibles.

Nous avons comparé cette méthode oracle avec les algorithmes naïf et optimisé par la géométrie dans [Aveneau 2003a]. Deux des scènes de tests sont illustrées sur la figure 5.10. La scène B contient 1352 faces, 1956 dièdres, et 8983 récepteurs. Par exemple, avec 2 réflexions et 0 diffraction, il faut 218 minutes en version naïve (pour plus de 16 milliards de



(a) Scene B

(b) Scene C

FIGURE 5.10 – Scènes de test pour la méthode oracle.

chemins à tester), 51 minutes en version géométrique et 6 minutes en version oracle.

La scène C contient 4038 faces, 5999 dièdres et 8204 récepteurs. Avec une version naïve il faudrait plusieurs milliers d'heures de calcul pour 2 réflexions et une diffraction (environ 2.40×10^{15} chemins), et plusieurs centaines en version géométrique. Avec l'oracle, le calcul de couverture est effectué en 11 minutes seulement.

La méthode est donc particulièrement utile pour générer rapidement des zones de couverture. Elle a été utilisée pour fournir des simulations dans de nombreux contrats avec des industriels (France Télécom, Alstom, Thales, ...).

5.3.1.2 Vers un graphe de visibilité en dimension 3

Le paragraphe précédent introduit deux greffons de CRT, dont l'objectif est la réduction des temps de simulation. Ils se basent tous les deux sur la modélisation d'un algorithme naïf avec un arbre de chemins potentiels. Ces premiers travaux nous ont conduit à l'idée d'un graphe dans lequel la visibilité entre deux faces est garantie. Autrement dit et indépendamment d'une quelconque position d'émission, lorsque deux primitives parmi les faces et dièdres ne se voient pas, alors elles ne sont pas reliées par une arête.

Un tel graphe peut reposer sur trois types de calculs de visibilité : exact, agressif, et *conservatif*. Le second, nous l'avons vu, permet des temps de simulation particulièrement intéressants ; néanmoins, étant agressif il rate des chemins de propagation valides, et donc n'est pas utilisable comme une méthode de référence. Nous avons écarté les méthodes conservatives, car elles risquent d'introduire des arêtes inutiles dans le graphe, et par conséquent de ne pas réduire suffisamment les temps de simulation.

Nous avons donc proposé, dans le cadre du Contrat de Recherche Externe n°42.538.233, avec France Télécom [Aveneau 2003b, Aveneau 2004c, Aveneau 2004d, Aveneau 2004b, Aveneau 2005], d'utiliser des méthodes exactes. Ce contrat nous a permis de financer le

doctorat de F. Mora. Nous avons fourni à France Télécom différents algorithmes de calcul de visibilité, y compris en version exacte. Pour une utilisation dans un graphe de visibilité, l'un des points importants concerne la visibilité entre une arête et soit une face soit une autre arête [Mora 2005b]. Nous avons montré qu'il est possible d'extraire cette information directement de la représentation explicite des polytopes résultant d'un calcul de visibilité, tel que celui proposé dans le §3.1.2.

5.3.1.3 Simulateur RaPSor

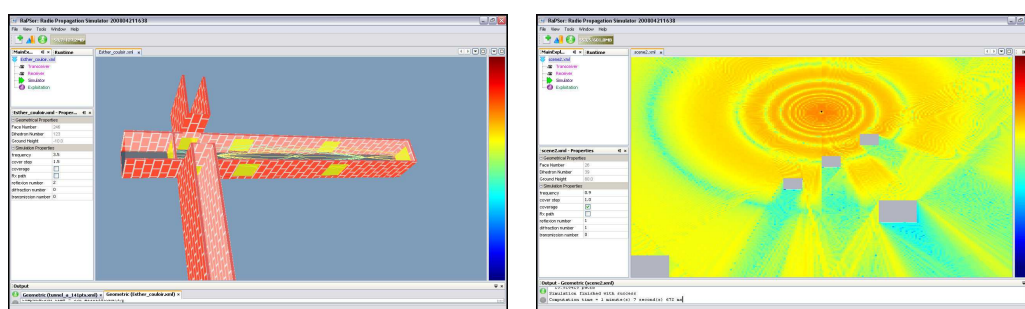
La diffusion du premier simulateur, à la fois pour l'enseignement et la recherche, nous a conduit à redéfinir un nouvel outil. En effet, nous étions confrontés à plusieurs problèmes : la portabilité, la maintenance, l'extensibilité.

Notre premier outil était écrit en C, avec une interface basée sur la boîte à outils GTK+. Les installations sur les différentes plateformes Windows, Linux et Mac OSX nécessitaient presque toujours une compilation dédiée, source de perte de temps.

Chaque mise à jour, soit après une correctif, soit après l'ajout d'une nouvelle fonctionnalité, impliquaient aussi une nouvelle compilation sur chaque poste utilisateur.

Enfin, chaque nouvel étudiant de DEA ou de Master recherche, et chaque thésard, dont l'objectif était l'ajout d'une nouvelle fonctionnalité, nécessitait mon aide pour au moins démarrer sinon réaliser le module.

Une nouvelle version était donc nécessaire afin de résoudre ces trois problèmes. Nous avons opté pour une version en Java (portabilité), basée sur la plateforme *Netbeans* (maintenance, via des mises à jour automatiques à chaque lancement de l'application). De plus, nous avons repensé le code afin de l'ouvrir un maximum (extensibilité, avec les patrons de conception Polymorphisme, Fabrique, Singleton et Injection de dépendance).



(a) Simulation indoor.

(b) Simulation outdoor.

FIGURE 5.11 – Exemple d'utilisation de RaPSor.

Notre second simulateur est nommé RaPSor. Il reprend et étend les capacités de CRT, et fonctionne en outdoor ou en indoor, en large bande ou en bande étroite (cf. figure 5.11). Son noyau fonctionnel permet la simulation basée rayons, dans des environnements purement spéculaires comme en radio, ou bien généraux comme en acoustique ou en optique.

Cette nouvelle version est utilisée aussi bien en enseignement qu'en recherche, à l'Université de Poitiers, au LEOST à Lille, à Limoges au sein d'XLIM et à La Rochelle au L3i pour les collaborations actives. Les deux greffons "radars à travers les murs" et "simulation en tunnels", présentées dans les sections suivantes, sont ainsi basés sur RaPSor.

5.3.2 Radars à travers les murs

Cette section présente nos travaux publiés dans [Liebe 2010], consistant en des simulations en ultra large bande (ULB) sous RaPSor.

5.3.2.1 Objectifs

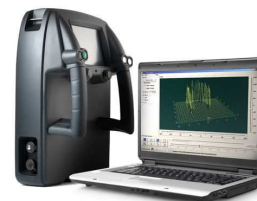
Une des applications de la technologie ULB est le radar permettant la "vision" à travers les murs. Différentes utilisations sont visées, aussi bien civiles (sauvetage de personnes enfouies ou lors d'incendie) que militaires (recherche de bombe, neutralisation, sauvetage). Ces radars peuvent capter des informations à travers des murs en béton (jusqu'à 5 GHz), en brique (jusqu'à 10 GHz) ou des cloisons (jusqu'à 60 GHz).



(a) Radar SAR ULB



(b) SoldierVision



(c) Autre radar portable

FIGURE 5.12 – Exemples de radars ULB.

Le fonctionnement d'un radar ULB est grosso modo le suivant : il émet sur une large bande de fréquences (d'au moins 1.5 GHz) un signal très court (une impulsion), et écoute en parallèle ce qui revient sur la même bande de fréquences. La largeur de bande assure de traverser une très large gamme de matériaux, et donc de détecter des signatures particulières ou des mouvements. Un usage typique est la détection de mines enfouies (*cf.* figure 5.12(a), radar SAR ULB, XLIM), ou bien le *SoldierVision* détectant le mouvement de la respiration (*cf.* figure 5.12(b) et figure 5.12(c)).

La figure 5.13 propose un exemple d'acquisition à travers un mur en béton. La scène contient une personne en mouvement (figure 5.13(a)). Deux acquisitions sont effectuées, l'une à un instant t_0 (figure 5.13(b)), l'autre 500ms plus tard (figure 5.13(c)). La figure 5.13(d) montre

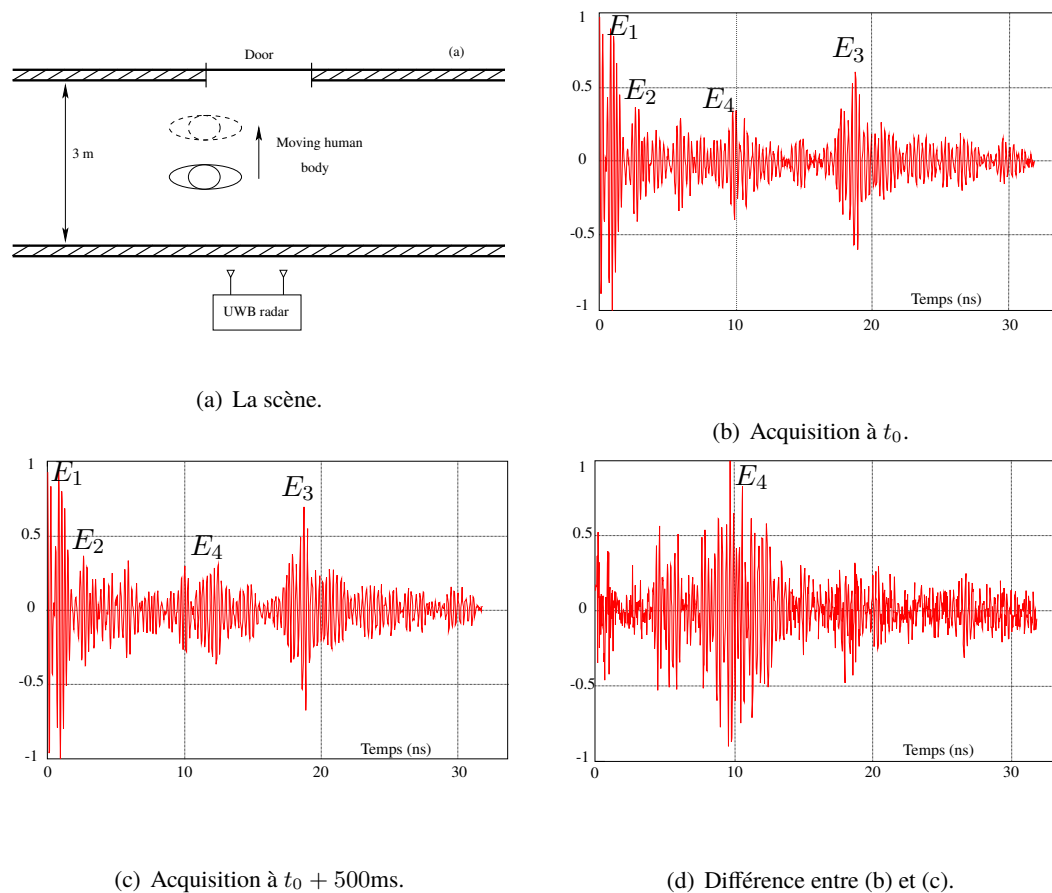


FIGURE 5.13 – Exemple d’une acquisition réelle. L’abscisse dans les figures (b) à (d) représente le temps de propagation en nano-seconde, et l’ordonnée une amplitude normalisée.

la différence entre ces deux acquisitions ; l’information relative à des éléments statiques de la scène est ainsi supprimée (échos E_1 , E_2 et E_3), et donc la signature de la personne en mouvement devient perceptible (l’écho E_4). Si les différents échos sont perceptibles, leur origine est difficilement compréhensible sans l’information détaillée de la scène. Or, justement, le but de ce genre de radar est de reconstruire l’environnement et de détecter d’éventuelles personnes.

Un approche purement expérimentale ne suffit pas à cataloguer tous ces échos. Pour permettre la conception et la validation d’un radar ULB, il est ainsi nécessaire d’augmenter la connaissance de la caractérisation des signaux ULB. Notre travail se place dans ce contexte, et vise à simuler le comportement d’un radar ULB, conçu par le L3I de La Rochelle.

Le diagramme de l’antenne ULB du radar rochelais est fourni par le L3I. Notre travail fût donc d’inclure ce système dans RaPSor, de définir des scénarios de simulation, et de les valider par comparaison à des mesures expérimentales.

5.3.2.2 Simulation ULB

Pour réaliser le greffon ULB, nous avons dû résoudre deux principaux problèmes. Le premier consiste à gérer l'ULB, et le second à modéliser correctement la transmission à travers des parois plus ou moins complexes.

La caractéristique principale d'un système ULB est sa bande de fréquences, au moins supérieure à 1.5Ghz. Une solution similaire à celle utilisée en synthèse d'images est d'associer à chaque chemin un vecteur de champs (ici électromagnétiques) pour quelques longueurs d'onde bien choisies. Mais cette approche ne convient pas, car le résultat attendu n'est pas un calcul en puissance, mais une réponse impulsionnelle caractérisant le canal de propagation. Il est nécessaire d'avoir un calcul "fin" pour la bande de fréquences. Utiliser un vecteur contenant quelques dizaines de milliers de champs pour des longueurs d'onde différentes engendrerait une consommation mémoire prohibitive³.

Nous utilisons donc un mécanisme différent : une première étape réalise la simulation géométriquement. Le calcul n'est pas physique, comme habituellement : ici nous stockons uniquement les chemins en visibilité, ainsi que tous les paramètres nécessaires à un calcul physique. Ces derniers sont multiples : orientation des rayons, propriétés des matériaux pour la réflexion, la diffraction et bien sûr la réfraction, angle d'incidence, normale, etc. Dans une seconde étape, chaque chemin valide est utilisé pour calculer une réponse impulsionnelle par longueur d'onde, ajoutée à la réponse globale du canal.

Considérant la réponse fréquentielle $H(f)$ en bande large du canal, et le signal émis $S(f)$ lui-aussi en version fréquentielle, alors la réponse fréquentielle en bande large est le produit suivant :

$$R(f) = S(f) \cdot H(f).$$

La réponse temporelle $r(t)$ du canal est alors la transformée de Fourier inverse de la réponse fréquentielle. Un exemple d'une telle transformation est proposé dans la figure 5.14.

5.3.2.3 Résultats

Dans [Liebe 2010], les résultats de simulation sont comparés à des mesures expérimentales. L'objectif est, rappelons-le, de mieux comprendre le canal de transmission, afin de construire un radar ULB de qualité.

Nous avons effectué deux séries de simulations. Dans la première, l'objectif est de vérifier la possibilité de retrouver et interpréter correctement une information pertinente dans la réponse impulsionnelle du canal $h(t)$ (obtenue par transformée de Fourier inverse de $H(f)$). La scène étudiée correspond à une scène réelle, dans laquelle des mesures expérimentales sont conduites. Il s'agit d'une simple pièce rectangulaire, contenant l'émetteur et le récepteur (cf. figure 5.15(a)). La simulation permet de retrouver l'ensemble des trajets principaux (cf. figure 5.15(b)). Cette étude préliminaire nous a notamment permis de paramétrer le nombre optimal d'interactions d'une simulation ULB. D'autres comparaisons

3. Dans RaPSor, la partie simulation est décorrélée de la partie exploitation d'une simulation.

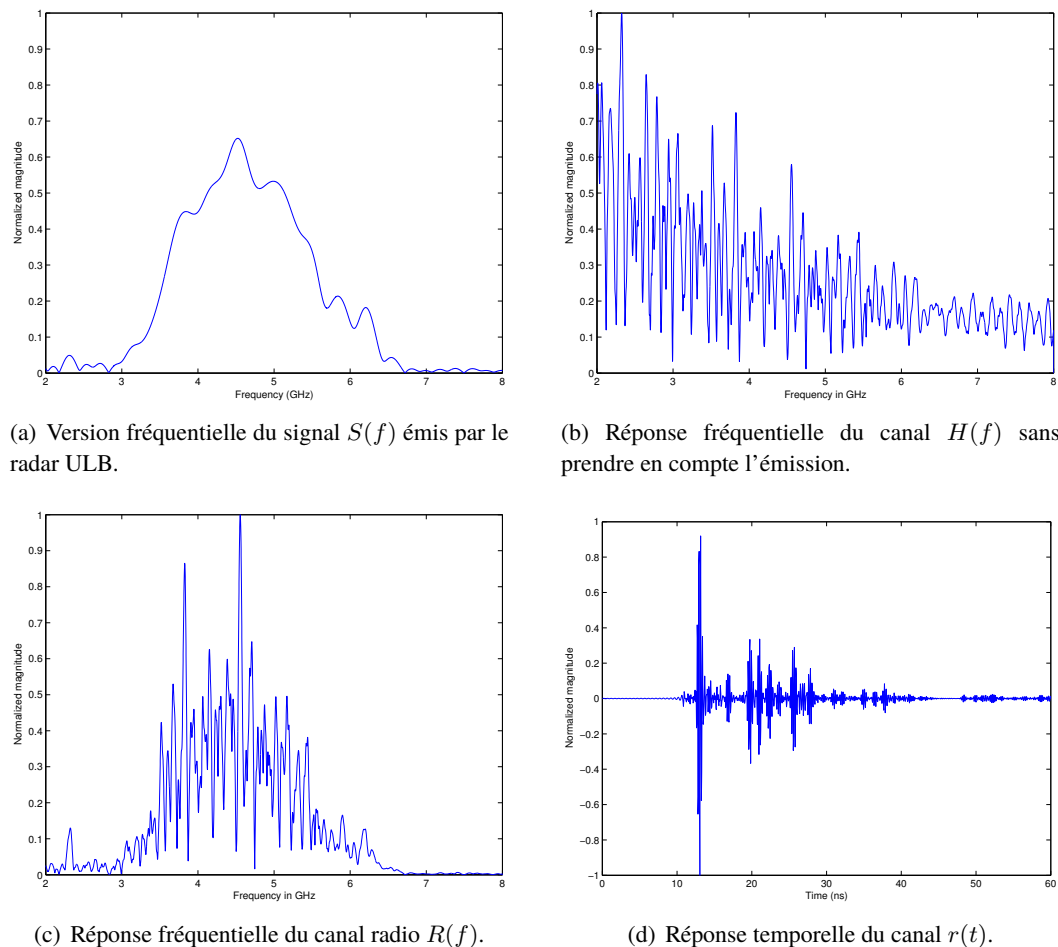
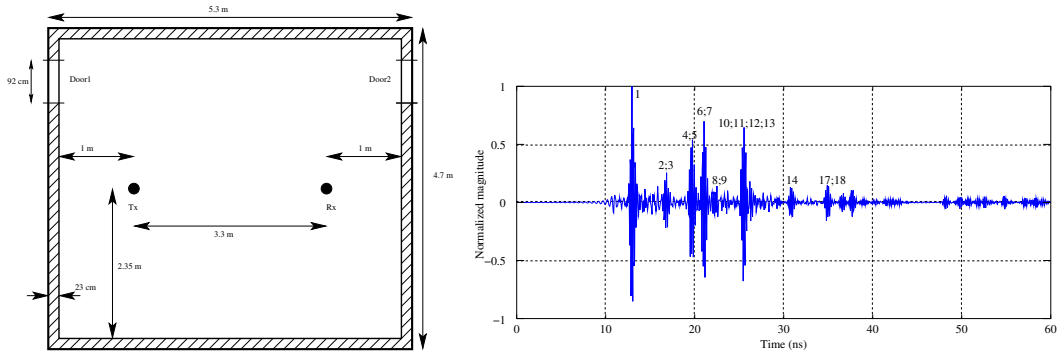


FIGURE 5.14 – Calcul de la réponse temporelle du canal de propagation : (a) le signal $S(f)$ émis en version fréquentielle ; (b) la réponse $H(f)$ en large bande du canal simulé ; (c) le produit $R(f) = S(f) \cdot H(f)$; (d) la réponse temporelle du canal.

furent conduites entre des mesures et de la simulation pour des positions d'antennes différentes. La conclusion est qu'une simulation utilisant 3 réflexions et 0 diffraction permet d'obtenir une réponse impulsionnelle de qualité suffisante pour retrouver et analyser l'ensemble des chemins de propagation.

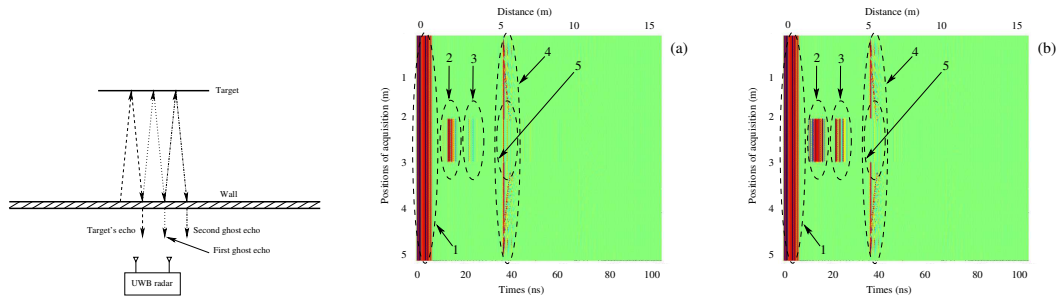
L'étape suivante est l'analyse d'une réponse temporelle du canal $r(t)$. La figure 5.14(d) correspond à la configuration de la scène de la figure 5.15. Il est relativement difficile de comprendre l'origine des différents pics. En étudiant la réponse impulsionnelle du canal, chaque trajet et donc chaque pic correspond à un chemin particulier. Cette étude nous a permis de mieux comprendre et donc interpréter une réponse temporelle d'un canal, autant en version simulée que mesurée.

La seconde série de simulation concerne un environnement différent, présenté sur la figure 5.16. L'objectif est ici d'analyser les échos fantômes, engendrés en pratique par des



(a) Plan de la pièce utilisée pour les mesures. (b) Simulation sous RaPSor, et identification des trajets connus.

FIGURE 5.15 – Simulation d’un environnement réel, pour trouver les paramètres optimaux d’une simulation radar ULB.



(a) Plan de la pièce utilisée pour la seconde série de mesures. (b) Image radar simulée pour un obstacle métallique. (c) Image radar simulée pour un obstacle diélectrique.

FIGURE 5.16 – Simulation de l’imagerie radar pour une même configuration, en variant la nature de l’objet détecté.

réflexions multiples d’un signal réfléchi. Plus précisément, le signal reçu par le radar est celui qu’il émet ; il est donc nécessairement réfléchi entre temps par un objet quelconque. Par contre, ce signal est bien souvent réfléchi une seconde fois sur la paroi traversée qu’écoute” le radar. Ceci engendre une troisième réflexion sur le même objet, et donc un premier écho fantôme. Et ainsi de suite, jusqu’à ce que le phénomène ne soit plus détectable. La figure 5.16(a) décrit la scène virtuelle, composée d’une paroi et d’un objet. Le radar est déplacé contre la paroi par pas de 5 cm, sur un axe horizontal. La figure 5.16(b) montre l’image radar obtenue pour un objet métallique : l’abscisse correspond à l’étalement temporel, et l’ordonnée à la position d’acquisition. La couleur indique l’amplitude du champ mesuré, avec un échelle logarithmique classique. La figure 5.16(c) illustre l’acquisition pour un objet de nature diélectrique. L’onde émise s’y réfléchit plus franchement, et donc l’objet et ses échos y sont d’autant plus visibles. Notons que la paroi est elle-même composée de plusieurs interfaces, expliquant la réception d’un signal dès les premiers instants de la simulation.

5.3.3 La simulation en tunnels

La thèse d'Émilie Masson porte sur la propagation d'ondes radio dans des tunnels. Il s'agit d'une thèse Cifre, en lien avec Alstom. Elle a été soutenue en 2010.

5.3.3.1 Objectifs

Les tunnels sont des environnements de propagation particuliers, ressemblant à des guides d'onde. Une simulation correcte nécessite un grand nombre de réflexions sur la surface du tunnel. Lorsque les parois du tunnel sont planes, alors une simulation classique est possible, et donne de bons résultats comparés aux mesures expérimentales. Dans ce cas, nous dirons qu'un tunnel est à profil rectangulaire et à section rectiligne.



(a) Section rectiligne.

(b) Section courbe.

FIGURE 5.17 – Deux tunnels à profil circulaire, mais différents par leurs sections.

Néanmoins, la plupart des tunnels comportent des surfaces courbes : si le sol reste plat, les “murs” sont souvent des surfaces courbes. Dans ce cas, une portion de tunnel rectiligne (*cf.* figure 5.17(a)) a une surface cylindrique, tandis qu'une portion courbe (*cf.* figure 5.17(b)) possède une surface de type torique. En pratique nous distinguons deux paramètres dans la forme d'un tunnel :

- son profil, rectangulaire ou circulaire, qui est en quelque sorte la forme de la cavité creusée par le tunnelier ;
- sa section, rectiligne ou circulaire, correspondant à la trajectoire du tunnelier.

Un tunnel est généralement constitué d'un ensemble de portions rectilignes et courbes. Nous avons ainsi quatre types de portions de tunnels. Les tunnels sont construits comme une suite de portions de même profil, mais avec des longueurs et des rayons différents.

Paradoxalement, la visibilité dans un tunnel ne permet pas de réduire la complexité des calculs. En effet, et contrairement à des cavités naturelles, les tunnels ont des “trajectoires” assez simples. Ainsi, un graphe de visibilité a de grande chance d'avoir une très grande densité (autrement dit, chaque noeud du graphe risque d'avoir un nombre d'arcs proche du nombre total de noeuds du graphe).

La difficulté d'une simulation est alors double :

- Prendre en compte la surface d'un tunnel, qui peut être plane, cylindrique ou torique.
- Obtenir une simulation correcte et rapide, malgré le grand nombre d'interactions nécessaires.

Parmi les travaux antérieurs, les méthodes basées rayons sont celles qui sont les plus pertinentes à la simulation de la propagation dans des grands tunnels de formes complexes, en utilisant un modèle polygonal approché [Chen 1996, Torres 1999]. Dans [Wang 2006], Wang *et al.* utilisent une notion de tube de rayons ; malheureusement leur article manque de détails, et leur démarche n'est donc pas reproductible. Enfin, dans [Didascalou 2000, Didascalou 2001] Didascalou *et al.* proposent d'utiliser un lancé de rayons associé à une méthode de normalisation des rayons, dont l'objectif est le traitement de la réception multiple d'un même trajet.

En effet, avec une méthode de type lancé de rayons les rayons sont émis aléatoirement depuis la source, et rebondissent sur les parois du tunnel ; la réception n'est donc jamais exacte, puisque la probabilité qu'un rayon aléatoire passe par la position exacte du récepteur est nulle. La réception utilise alors une sphère de réception centrée sur le récepteur ; le rayon de cette sphère est variable, et dépend de la longueur du chemin depuis le point d'émission jusqu'au point le plus proche du récepteur. Avec des surfaces courbes, il arrive souvent que deux trajectoires proches ne représentent en réalité qu'un unique et même trajet ; or, la somme des deux champs associés aux trajets approchés correspond grosso modo au double du champ du trajet théorique. La normalisation consiste ainsi à identifier ce problème, et à le régler.

5.3.3.2 Apports

Nous avons étudié la propagation dans les tunnels en proposant deux démarches différentes, chacune étant validée par des mesures expérimentales conduites soit par le LEOST, soit par Alsthom.

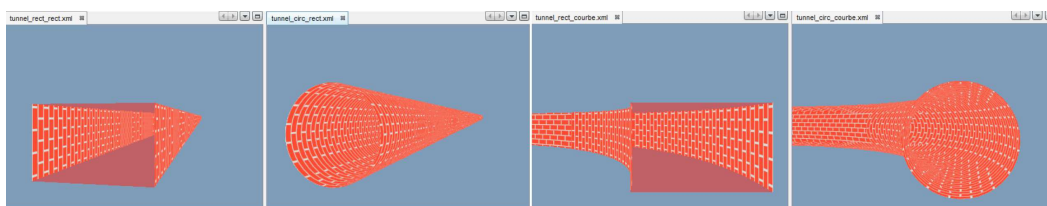


FIGURE 5.18 – Différentes sections de tunnel dans RaPSor, de gauche à droite : à section droite et profil rectangulaire, à section droite et profil circulaire, à section courbe et profil rectangulaire, et à section courbe et profil circulaire.

Notre premier greffon utilise les capacités de simulation de RaPSor classique, autrement dit la simulation exacte sur un ensemble de faces planes. Un tunnel à profil rectangulaire et à section courbe possède néanmoins des surfaces de type cylindrique (*cf.* figure 5.18). Ainsi, même pour un profil rectangulaire il est nécessaire de facettiser la surface du tunnel en un ensemble de polygones. Ce mécanisme est similaire à [Chen 1996, Torres 1999] et souffre

du même défaut : les calculs sont imprécis, en raison justement de la facettisation.

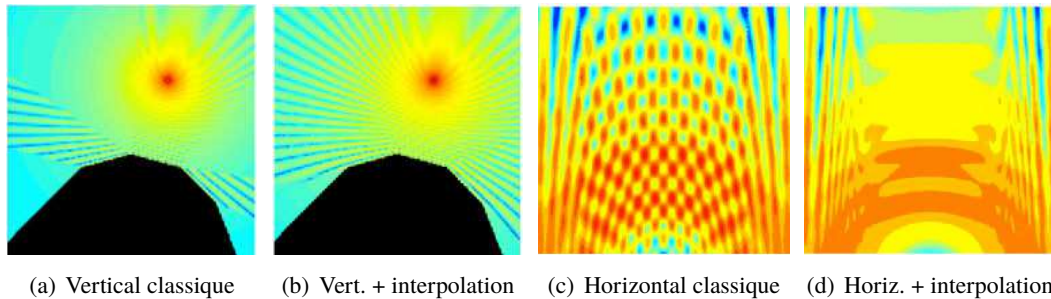


FIGURE 5.19 – Simulation sur modèle polygonaux, en version classique ou par lancé de rayons avec interpolation des normales.

Ce problème est visualisé sur la figure 5.19 avec un chemin simplement diffracté, et le trajet direct. La simulation y est effectuée dans une configuration concave, afin de mettre en évidence le problème de la facettisation ; bien entendu, il reste le même dans une configuration convexe. La somme des différentes contributions exactes engendre des interférences. Sur la figure 5.19(a), la simulation est effectuée dans un plan vertical ; sur la figure 5.19(c), elle l'est dans un plan horizontal. Clairement, ces résultats de simulation ne sont pas corrects.

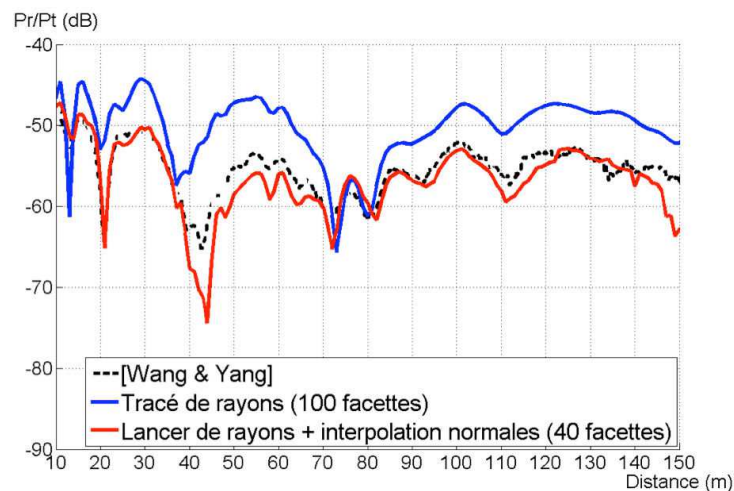


FIGURE 5.20 – Comparaison des méthodes classique et lancé de rayons avec la littérature [Wang 2006].

Plutôt qu'un mécanisme de simulation classique, nous proposons d'utiliser un lancé de rayons. Les tunnels sont toujours modélisés à l'aide de polygones, mais le calcul des chemins réfléchis utilise une normale interpolée, ce qui permet de couvrir tout l'espace [Masson 2010]. Les figures 5.19(b) et 5.19(d) illustrent la correction apportée. Nous avons vérifié la convergence des puissances calculées pour un ensemble de récepteurs positionnés dans le tunnel par comparaison avec [Wang 2006], comme illustré dans la figure 5.20.

Si cette première solution est satisfaisante pour un tunnel à profil circulaire, elle ne donne

pas de bons résultats dans le cas d'un tunnel à section courbe. Notamment, les calculs physiques nécessitent eux aussi la prise en compte de la courbure de la surface de réflexion, qui induit celle des ondes propagées. De plus, la courbure implique géométriquement des difficultés pour discriminer les rayons au moment de la réception : en lancé de rayons pour la radio, la réception est assurée par un test de pénétration des rayons avec une sphère centrée autour du point de réception ; les courbures induisent des concentrations ou des divergences, rendant les méthodes usuelles d'émission et de réception inopérantes [Didascalou 2001].

Nous avons proposé une seconde méthode de simulation reposant sur des surfaces courbes de types cylindre et tore. Chaque chemin est tracé comme dans un lancé de rayons en synthèse d'images. Notre apport consiste à réduire les problèmes en lien avec les trajets multiples. L'émission est générée par une méthode de Hammersley à faible discrétance [Lemieux 2009], qui d'après nos expériences fournit les meilleurs résultats notamment en minimisant les problèmes de réceptions multiples. La réception est effectuée par test de pénétration d'une sphère dont le diamètre est déduit de l'écart maximal entre deux rayons émis. Pour traiter les réceptions multiples, chaque chemin pénétrant la sphère de réception est utilisé pour déterminer un chemin exact, par une méthode numérique de minimisation de distance (Levenberg-Marquardt). Les chemins reçus devenant exacts, il est facile d'éliminer les doublons en les insérant dans une liste triée par distance de propagation. La comparaison avec des mesures effectuées par le LEOST et par ALSTHOM a permis de valider la méthode [Masson 2011, Masson 2013]

5.4 La simulation pour l'optique sans fil

Les possibilités de commutation rapide des diodes électroluminescentes (ou LED), ainsi que leur utilisation de plus en plus généralisée pour l'éclairage domestique, permettent d'imaginer des réseaux de communication sans fil ne reposant pas sur la radio, et donc potentiellement moins craints par les utilisateurs (à tort ou à raison). Les applications sont multiples, notamment pour le monitoring de patients en milieu hospitalier. Si le Lifi, pour *Light Fidelity*, repose sur ce principe, il est pensé pour fonctionner principalement en visibilité directe, avec un débit théorique de 1 GBit/s par LED.

Notre approche est d'utiliser cette technologie en environnement quelconque, à bas débit. Le bas débit permet notamment d'éviter les interférences inter symboles tout en conservant du multitrajet, supprimant de fait la nécessité d'une visibilité directe.

Sur ce genre de réseau, il est important d'utiliser des protocoles de communication robustes. En collaboration avec le département C2S2 du laboratoire XLIM, nous travaillons sur la chaîne complète de transmission, de la simulation de propagation à la validation de nouveaux protocoles de communication pour les réseaux optiques sans fil.

5.4.1 Nouvelle équation de transfert lumineux

L'objectif de nos simulations optiques est la validation de protocoles réseaux. Il est donc nécessaire d'obtenir, comme en radio, une réponse impulsionnelle du canal. Ceci diffère du calcul d'une luminance reçue en un pixel dans le domaine du rendu de l'informatique graphique. Si la physique est la même, il faut néanmoins utiliser une équation de transport lumineux légèrement différente (*cf.* annexe A).

La réponse impulsionnelle prend la forme d'une courbe de puissance reçue en fonction du temps. Comme en radio, nous supposons l'émission d'une impulsion lumineuse. Le traitement numérique du signal lumineux nous permet en post-traitement de convoluer le signal par une impulsion réaliste (par exemple issue d'une mesure de LED).

L'émission de l'impulsion lumineuse est effectuée à un instant $t = 0$. Pour un récepteur donné, nous calculons la puissance reçue :

$$P(t) = \int_A E(x, t) dx,$$

où A est la surface du récepteur, et E est l'éclairement reçu sur la surface du récepteur à l'instant t :

$$E(x, t) = \int_{\Omega} L_i(x, t, \mathbf{i}) \cos \varphi_i d\mathbf{i},$$

où Ω est l'hémisphère supérieur centré en x . La luminance incidente $L_i(x, t, \mathbf{i})$ dépend donc du temps t . Cette luminance est celle émise en y , le point visible depuis x dans la direction \mathbf{i} , mais à un instant différent :

$$L_i(x, t, \mathbf{i}) = L_o \left(y, t - \frac{\|y - x\|^2}{c}, -\mathbf{i} \right),$$

où c est la vitesse de la lumière. Ainsi, l'ETL classique n'est pas exactement celle que nous allons utiliser dans nos simulations, mais une variante subtile⁴ reposant sur cette notion de temps de propagation :

$$L_o(x, t, \mathbf{o}) = L_e(x, t, \mathbf{o}) + \int_{\Omega} L_i(x, t, \mathbf{i}) f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\mathbf{i}. \quad (5.1)$$

La différence de la durée de propagation d'un trajet mise de côté, il apparaît vite que les méthodes de rendu classiques doivent être réutilisables presque directement. Les paragraphes suivants présentent deux études préliminaires sur la simulation optique par les méthodes de Monte-Carlo. Il s'agit des travaux effectués par des stagiaires de Master recherche, et principalement par Abdeslam Behloui en 2012, en lien avec Pierre Combeau du SIC et Anne Julien-Vergonjanne du C2S2.

5.4.2 Méthodes de Monte-Carlo

Cette section présente deux simulateurs optiques basés sur les méthodes de Monte-Carlo.

4. Au sens où le terme t joue un vrai rôle.

5.4.2.1 Monte-Carlo depuis les émetteurs

Le premier simulateur effectue une estimation de Monte-Carlo depuis les émetteurs. Nous partons en pratique de l'équation en potentiel de [Pattanaik 1995], formalisant l'action d'une source ponctuelle et placée en un point x , unitaire (émettant "1"), et directionnelle à l'extrême (n'émettant que dans une direction unique). Cette action est donnée pour un point y visible depuis x , en utilisant la notion de durée de propagation :

$$W(x, \mathbf{i}, t) = g(x, \mathbf{i}, t) + \int_{\Omega} f_r(y, \mathbf{i}, \mathbf{o}) W(y, \mathbf{o}, t) (\mathbf{n}_x \cdot \mathbf{o}) d\omega_{\mathbf{o}}, \quad (5.2)$$

dans laquelle Ω est l'hémisphère centré en y , \mathbf{i} est la direction $x \rightarrow y$, et la fonction $g(x, \mathbf{i})$ modélise une réception directe de l'action de la source potentielle en x sur un récepteur :

$$g(x, \mathbf{i}) = \begin{cases} 1, & \text{ssi } (x, \mathbf{i}) \text{ éclaire directement,} \\ 0, & \text{sinon.} \end{cases}$$

Cette expression de l'action potentielle d'une source sur un point y est duale de l'équation (5.1) de transfert lumineux temporelle. Une estimation de Monte-Carlo est :

$$\begin{aligned} \widehat{E}(t) &\approx \frac{1}{N} \sum_{k=1}^N \frac{L_o(x_o, 0, \mathbf{o}) |\mathbf{o} \cdot \mathbf{n}_o|}{p_o(\omega_o, x_o)} W(\widehat{x_o}, \mathbf{o}, t) \\ &\approx \frac{1}{N} \sum_{k=1}^N \left[\frac{L_o(x_o, 0, \mathbf{o}) |\mathbf{o} \cdot \mathbf{n}_o|}{p_o(\mathbf{o}, x_o)} \sum_{i=0}^{\infty} \left\{ \prod_{j=1}^i \frac{f_r(x_j, \mathbf{i}_j, \mathbf{o}_j) |\mathbf{i}_j \cdot \mathbf{n}_j|}{p_j(\mathbf{o}_j, x_j)} \right\} g(x_i, \mathbf{o}_i) \right], \end{aligned}$$

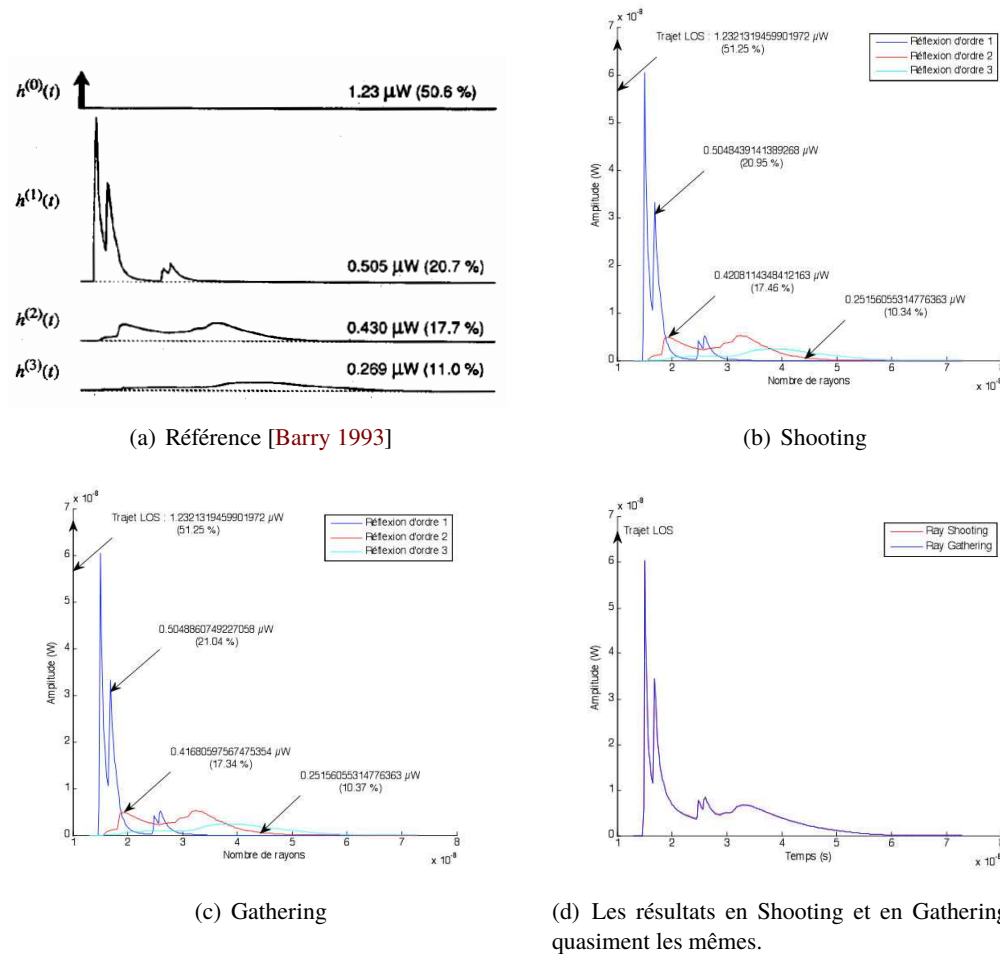
où $p_i(\mathbf{o}_i, x_i)$ est la probabilité du choix de la direction \mathbf{o}_i en un point x_i , les directions d'incidence sont opposées à celles d'émission, *i.e.* $\mathbf{i}_k = -\mathbf{o}_{k-1}$ pour $k > 0$.

Notre premier simulateur optique repose sur cette expression, en ajoutant pour réduire la variance à la fois un échantillonnage d'importance et l'estimation du prochain événement [Veach 1997], et en calculant des chemins de longueur donnée par l'utilisateur. Ce dernier point peut surprendre, mais est rendu nécessaire par la démarche de validation choisie, utilisant des résultats de simulation issus de la littérature et validés eux-mêmes par confrontation à la mesure [Barry 1993].

5.4.2.2 Monte-Carlo depuis les récepteurs, validation

Partant de l'équation temporelle de transfert lumineux 5.1, nous proposons d'effectuer une estimation depuis les récepteurs. Nous utilisons une équation similaire à celle de l'estimation depuis les sources. Ce second simulateur repose comme le premier sur l'échantillonnage d'importance et l'estimation du prochain événement.

La figure 5.21 propose les réponses impulsionnelles calculées sur une scène extraite de la littérature [Barry 1993]. La scène est un simple parallélépipède, la source est directionnelle, et les résultats de Barry *et al.* sont obtenus par une méthode de radiosité un peu particulière (le maillage étant de plus en plus grossier à mesure que le nombre de réflexions augmente).

FIGURE 5.21 – Résultats de Barry *et al.*, et nos deux méthodes.

Les résultats de référence sont donnés dans la figure 5.21(a). Les figures 5.21(b) et 5.21(c) illustrent les résultats de nos simulateurs depuis les sources et depuis les récepteurs. Enfin, la figure 5.21(d) montre que les deux simulateurs fournissent des réponses impulsionnelles presque identiques (pour un million d'échantillons, ce qui explique les légères différences).

Une étude de la vitesse de convergence a montré que l'estimation de Monte-Carlo depuis les récepteurs converge plus rapidement que celle depuis les sources. Néanmoins, en fonction des besoins de simulations les deux modes sont plus ou moins pertinents : par exemple en mode SIMO⁵, l'estimation depuis les sources permet de calculer plusieurs réponses impulsionnelles plus rapidement que celle depuis les récepteurs.

5. *Single Input, Multiple Output*, un émetteur et plusieurs récepteurs.

5.5 Conclusion

Nos travaux sur la visibilité trouvent leurs racines dans nos contributions sur la simulation de propagation d'ondes. En effet, la complexité d'une simulation en environnement purement spéculaire rend rapidement impraticable les solutions dites exactes. La réduction de la complexité passe par un *élagage* de l'arbre des chemins potentiels. Utiliser un graphe de visibilité exact est un mécanisme théoriquement simple pour réduire cette complexité.

Nos travaux sur la simulation de propagation d'ondes sont relativement appliqués, notamment afin de fournir des simulations et des outils à nos collègues électroniciens. Nous avons donc proposé deux outils, l'un déposé à l'AFUL, l'autre disponible en licence libre sur *sourceforge*, permettant d'effectuer des simulations. Le premier ne permet que des simulations dans le domaine des ondes radio. Il a été utilisé dans le cadre de la thèse de Pierre Combeau, soutenue en 2004, à laquelle j'étais associé comme encadrant. Il a aussi été utilisé par d'autres thésards du laboratoire SIC (F. Escarieu, S. Reynaud, Y. Cocheril, parmi ceux avec qui j'ai directement collaboré), ainsi que dans des contrats avec des industriels (pour effectuer des simulations).

Notre second simulateur, RaPSor, a été développé afin de permettre une utilisation plus souple, notamment pour permettre des développements ultérieurs autonomes. Il est ainsi ouvert et extensible. De nombreux travaux reposent dessus, au SIC, à XLIM, au L3I, au LEOST, et il a été diffusé internationalement (notamment auprès d'étudiants en Allemagne, Iran et Égypte). Il a entre autre été utilisé dans le cadre de la thèse d'Émilie Masson, soutenue en 2010.

Ce simulateur permet d'utiliser différentes physiques, autres que l'électromagnétisme, telles que l'acoustique ou l'optique. Nous avons notamment démarré des travaux sur la simulation optique pour la caractérisation du canal de propagation en optique sans fil. Ainsi, Abdeslam Behlouli va débiter un doctorat en Novembre 2013 sur la simulation optique en environnements dynamiques, avec un encadrement conjoint avec Pierre Combeau du département SIC de l'institut XLIM.

Conclusion & Perspectives

Sommaire

6.1 Conclusion	123
6.1.1 Modèles pour le calcul de visibilité	123
6.1.2 Synthèse d'images	124
6.1.3 Simulation de propagation d'ondes	125
6.2 Programme de recherche	127
6.2.1 Aspects formels	127
6.2.2 Perspectives en rendu	129
6.2.3 Perspectives en simulation de propagation d'ondes	132

6.1 Conclusion

Cette section résume les travaux présentés dans ce mémoire.

6.1.1 Modèles pour le calcul de visibilité

Le passage à la dimension 3 ne permet pas une adaptation immédiate des méthodes de calcul de visibilité fonctionnant en dimension 2. La conséquence est que les travaux antérieurs sur la visibilité en dimension 3 sont principalement orientés sur l'efficacité des algorithmes proposés, au détriment de la qualité du résultat. Ainsi, ils sont uniquement utilisés pour des applications de la visibilité très tolérantes quant au traitement des erreurs. Ils souffrent en réalité d'un manque de réflexion sur les problèmes liés à ce changement de dimension, et plus généralement à la nature des droites dans un espace géométrique \mathcal{G}_n .

Partant de ce constat, nous avons utilisé l'algèbre de Grassmann pour formaliser un espace de droites valide en dimension $n \geq 2$. Le produit de Grassmann permet de représenter la dépendance linéaire entre des sous-espaces plats de l'espace géométrique. Ainsi, une droite est-elle simplement le produit de deux de ses points distincts. Cette définition particulièrement simple repose sur un formalisme complet autorisant des démonstrations formelles. La conséquence immédiate est une meilleure compréhension des travaux précédents et notamment de leurs lacunes. Mieux, cette formalisation permet d'entrevoir des améliorations possibles des travaux précédents, et surtout d'unifier l'ensemble des méthodes quelle que soit la dimension de l'espace \mathcal{G}_n . En effet, notre démarche formelle permet une réflexion indépendante de la dimension de l'espace géométrique.

Cette approche nous a permis de proposer un premier *modèle pour le calcul de la visibilité exacte*, dans le cadre de la thèse de Sylvain Charneau (2007). Dans ce modèle, nous résolvons les problèmes des méthodes précédentes, comme la nature de l'opérateur *side*, le problème du polytope minimal, *etc.* Nous avons ainsi proposé un calcul de visibilité fonctionnel, permettant de représenter l'ensemble des droites traversant deux faces quelconques, mais sans traverser aucun obstacle. En utilisant les silhouettes des objets bloquants, cet ensemble est proche d'être minimal. Notre modèle représente la visibilité en utilisant un arbre binaire initialement proposé durant la thèse de Frédéric Mora (2006), et rendant possible son exploitation ultérieure.

Un tel calcul améliore les méthodes précédentes et permet de dépasser de loin leurs limites. Néanmoins, la visibilité est un problème ardu, dont la complexité est en théorie $\mathcal{O}(n^4 \log n)$. Ceci limite la taille des scènes exploitables, dans le sens où nos arbres de visibilité peuvent devenir particulièrement gros. Afin d'exploiter la visibilité sur des scènes complexes, nous avons avec Frédéric Mora proposé un second *modèle reposant sur un calcul à la demande*. L'arbre de visibilité n'est calculé que dans les parties nécessaires à son exploitation, lorsqu'un besoin existe. En exploitant l'arbre uniquement dans l'espace géométrique \mathcal{G}_n , une grande partie de l'information n'est jamais requise, et donc calculée. Cette seconde approche permet donc d'augmenter considérablement le nombre d'occulteurs présents entre deux faces lors d'un calcul de visibilité.

Ces deux modèles sont utilisables dans toutes applications nécessitant des calculs de visibilité, peu importe la dimension du problème : pour des points, des arêtes, des faces, des volumes dans \mathcal{G}_3 , et pour des k -faces dans \mathcal{G}_n . L'arbre binaire de visibilité contient une information booléenne de visibilité dans les feuilles, et des hyperplans correspondant aux bords des occulteurs dans les nœuds internes.

6.1.2 Synthèse d'images

Une partie de nos travaux en synthèse d'images utilisent nos modèles de visibilité, à commencer par le rendu d'ombres douces. Ce problème est intrinsèquement de nature point-à-face. Nous l'avons exploré dans le cadre des deux thèses de F. Mora et S. Charneau. Nous utilisons néanmoins des calculs de visibilité entre deux faces, calculs rentabilisés ensuite en multipliant leurs utilisations. L'une des faces est la source lumineuse, l'autre est une face géométrique de la scène. L'information de visibilité stockée dans un arbre est ainsi exploitée en plusieurs positions de calcul données sur la seconde face. Cela permet de calculer l'occultation d'une source lumineuse beaucoup plus rapidement qu'avec un lancé de rayons traditionnel par exemple. En outre, le résultat est de qualité au sens où l'information de visibilité est exacte.

Une seconde application de nos calculs de visibilité concerne le rendu de l'occlusion ambiante. L'information de visibilité nécessite alors une notion de profondeur : ce n'est plus simplement un problème d'occultation. A cette fin, l'arbre de visibilité est modifié afin d'inclure la profondeur de l'occulteur. Une feuille contient une information de visibilité ou bien la plus proche face bloquant la visibilité. Cet arbre permet, en version paresseuse, d'ef-

fectuer des rendus de qualité et en des temps bien inférieurs à ceux des méthodes basées rayons.

Enfin, une troisième application de nos modèles de visibilité est en cours de soumission en revue. Elle se place dans le contexte de la thèse de Romuald Perrot (2012). La visibilité est utilisée lors de l'estimation de densité par noyau dans une carte de photons (par exemple). Ces méthodes ont un faible biais théorique lorsque le calcul est effectué loin des frontières du domaine du noyau d'estimation. En pratique, l'estimation est souvent à proximité d'une frontière du domaine, et donc un fort biais apparaît. Nous avons montré que le domaine peut-être calculé de façon exacte en utilisant la visibilité. Dès lors, il est possible de normaliser le noyau d'estimation pour contrôler le biais dans des proportions raisonnables.

Nous avons aussi proposé deux autres applications ne reposant pas sur nos modèles de visibilité. La première, toujours dans le cadre de la thèse de Romuald Perrot, concerne une réécriture de l'équation de rendu en remplaçant ses termes par des lobes de cosinus. Les lobes forment une base infinie de fonctions, et remplacent relativement simplement les fonctions de réflectance. La seule difficulté réside dans le traitement de la visibilité. Nous avons néanmoins proposé un traitement efficace de la visibilité permettant de reconstruire l'information uniquement avec des lobes de cosinus. Notre démarche est validée avec une application interactive de rendu, avec des ombres douces ou quasiment dures, et différents types de matériaux.

Enfin, dans le cadre de la thèse de Maxime Maria, démarrée en octobre 2012, nous avons proposé une structure accélératrice permettant le rendu interactif dans des grands bâtiments. Elle est automatiquement générée par le modèleur architectural basé sur le modèleur topologique *Moka*. Notre structure est de type cellules et portails, et comporte deux améliorations principales par rapport aux travaux antérieurs. D'une part, les cellules collent parfaitement à la géométrie des bâtiments, dans le sens où les bords d'une cellule sont soit des portails, soit des polygones du bâtiments (murs, sols, plafonds, *etc.*). D'autre part, le parcours de notre structure est optimisé en tenant compte des relations topologiques entre les k -faces en toute dimension (pour tout k), et non pas seulement pour $k = 3$ comme dans les travaux antérieurs (liens entre volumes). Ainsi, notre structure permet un calcul très rapide autant de la visibilité que de l'occultation depuis un point, pour lancer des rayons ou pour calculer l'éclairage des sources. Le rendu est alors interactif aussi bien sur CPU que sur GPU.

6.1.3 Simulation de propagation d'ondes

La caractéristique d'une simulation de propagation radio réside dans les phénomènes électromagnétiques simulés : la diffraction, les réflexion et réfraction purement spéculaires. Ces phénomènes reposent sur le principe de Fermat étendu, stipulant que la durée de propagation le long d'un chemin est stationnaire. Une simulation exacte consiste à calculer l'ensemble *déterministe* des chemins menant de l'émetteur au récepteur. Le calcul est d'une complexité voisine de $\mathcal{O}(n^i)$, pour n polygones et i interactions. Une simulation naïve va tester tous les chemins possibles, et vérifier leurs existences réelles. Nos optimisations naissent d'une organisation de ces chemins virtuels dans un graphe. Notons que dans cette notion de *graphe*

de visibilité réside l'origine de nos travaux sur la visibilité.

Notre premier simulateur radio, *CRT*, a été utilisé au laboratoire SIC et en enseignement à l'Université de Poitiers. Nous avons présenté trois greffons de *CRT* en rapport avec la visibilité. En parcourant le graphe des chemins potentiels, le premier propose de vérifier leur existence géométrique. Le second repose sur une évaluation agressive de la visibilité, et effectue des simulations particulièrement rapidement. Enfin un troisième greffon vise la simulation de grands environnements urbains. Dans le cadre de la thèse de Pierre Combeau (2004), cette méthode repose sur un lancé de faisceaux en dimension 2. La construction des faisceaux est optimisée via la géométrie discrète, et les faisceaux sont stockés dans un graphe afin d'autoriser de multiples diffractions.

Notre premier simulateur a été très utilisé dans les années 2000 à 2006, succès qui a fini paradoxalement par poser des problèmes pour la maintenance, les mises à jour et l'ajout de nouveaux codes par des étudiants ou collègues. J'ai donc décidé de développer un nouveau simulateur, qui soit à la fois ouvert et extensible. Cette seconde application est *RaPSor*. Elle est portable car réalisée en Java, et permet des mises à jour automatiques car basée sur la plate-forme Netbeans. De plus, elle est conçue pour permettre d'ajouter facilement des greffons, aussi bien pour la simulation proprement dite, mais aussi pour utiliser les résultats d'une simulation, pour définir de nouveaux formats de fichiers, et pour ajouter des nouvelles physiques pour la simulation acoustique, radio ou optique.

Sur la base de *RaPSor*, nous avons participé au développement d'un greffon permettant de valider par des simulations un modèle d'antenne pour la vision à travers les murs. Cette antenne est un radar à *Ultra Large Bande*, permettant par exemple de détecter des mouvements ou encore la présence d'une forme métallique enfouie dans le sol. Notre simulation de radar ULB repose sur un premier greffon simulant notamment la transmission à travers des murs composés de multiples couches. Le second greffon exploite les résultats géométriques de la simulation, et applique les calculs physiques pour chaque chemin et pour chaque longueur d'onde. En utilisant ces deux greffons dans *RaPSor*, nos collègues du L3I à La Rochelle ont amélioré leur connaissance du canal ULB, afin d'améliorer leur modèle de radar ULB.

Une seconde application de *RaPSor* concernent la simulation radio dans les tunnels courbes, dans le cadre de la thèse Cifre d'Émilie Masson (2010), en lien avec Alstom et le LEOST à Lille. Nous avons proposé un greffon pour les tunnels facettisés, en utilisant un lancé de rayons et une correction des longueurs reposant sur une interpolation des normales. Pour des tunnels plus réalistes, nous avons créé un second greffon utilisant des surfaces courbes. La difficulté résidait principalement dans la détection des trajets multiples, reçus à tort en plusieurs exemplaires. En utilisant une méthode numérique pour minimiser les trajectoires des chemins, notre greffon offre une détection robuste de ces multiples réceptions.

Basé sur la propagation le long de rayons, *RaPSor* permet d'utiliser n'importe quel type de physique reposant sur cette notion de "rayon". Nous l'utilisons depuis peu pour effectuer des simulations de propagation en optique sans fil bas débit. Les simulations reposent sur des méthodes de Monte-Carlo, et permettent de calculer la réponse impulsionnelle du canal de propagation (la puissance reçue étalée dans le temps). Nos premiers résultats sont très encourageants, notamment en comparant nos simulations à des travaux antérieurs. Cette

thématique se poursuit avec la thèse de Abdeslam Behloui qui démarre fin Octobre 2013.

6.2 Programme de recherche

Les activités de recherche résumées ci-dessus comportent différents aspects pouvant être classés en trois catégories :

1. Les aspects formels pour la modélisation des problèmes. Avec les algèbres géométriques, nous avons unifié les méthodes de visibilité et obtenu des réponses à des problèmes ouverts en dimensions 3 et supérieures ; plus modestement, nous avons aussi proposé de nouveaux modèles tels que les lobes de cosinus, et travaillé sur différents aspects des estimations de densité et des méthodes de Monte-Carlo.
2. Des aspects algorithmiques, comme les arbres de visibilité en versions exacte et paresseuse, les optimisations basées sur la visibilité pour la simulation de propagation, la visibilité binaire ou avec profondeur, etc.
3. Enfin, et en liaison avec les précédents, des aspects applicatifs, orientés autour de l'image (ombres, *ambient occlusion*, illumination globale) et de la propagation d'ondes (radio en 2 et 3 dimensions, radar ULB, optique sans fil).

Le programme de recherche proposé repose sur cette démarche qui nous semble essentielle. Néanmoins sa présentation est structurée légèrement différemment, compte tenu de l'imbrication étroite des 2 derniers aspects.

6.2.1 Aspects formels

Concernant les modèles, notre programme est focalisé sur l'utilisation des algèbres géométriques. Nous avons détaillé dans le chapitre 3 l'algèbre de Grassmann, qui en est un cas particulier. Les algèbres géométriques reposent sur le produit géométrique, qui englobe à la fois le produit de Grassmann et le produit scalaire :

$$ab = a \wedge b + a \cdot b.$$

Le produit de Grassmann fournit toujours un résultat sur $\bigwedge^k(\mathbb{R}^n)$, pour $k \geq 1$, alors que le codomaine du produit scalaire est \mathbb{R} . Ainsi le produit géométrique contient une information plus complète que les deux autres. Il permet notamment de représenter des objets géométriques tels que les cercles ou les sphères, via le *modèle conforme*. Ce dernier est construit sur $\mathbb{R}^{4,1}$, un espace de signature $(4, 1, 0)$ ayant donc 4 vecteurs de carré positif, 1 de carré négatif et 0 de carré nul. L'un des atouts principaux du modèle conforme réside en ce qu'il manipule des objets de l'espace Euclidien, en conservant les notions de distance. Il permet aussi de représenter via les *rotors*¹ l'ensemble des transformations affines usuelles.

1. Dans les algèbres géométriques, un *rotateur* ou rotor en anglais est le produit géométrique de deux 1-vecteurs. Cette notion permet notamment de définir les complexes en dimension n , quaternions et octonions inclus, de façon particulièrement élégante.

Projet à court terme. Nous proposons de commencer par étudier la notion d'orientation dans le modèle conforme $\mathbb{R}^{4,1}$. En effet, les calculs de visibilité sont bâtis sur l'orientation entre une droite et une $(n-2)$ -face. L'espace des 2-vecteurs n'est pas, comme avec l'algèbre de Grassmann, l'espace des droites mais celui des 2-points, un concept propre au modèle conforme. Dans $\mathbb{R}^{4,1}$ les droites sont plongées dans l'espace des 3-vecteurs, contenant des cercles (une droite étant un cercle particulier ayant un point à l'infini). De plus, les plans partagent avec les sphères l'espace des 4-vecteurs (là encore, un plan n'est qu'une sphère ayant un point à l'infini). Notre idée repose sur l'utilisation de l'orientation entre un cercle ou une droite et le bord d'un objet ou encore un objet lui-même [Tenorio 2011, Aveneau 2012]. Cette étude théorique va se poursuivre avec Angel Tenorio, de l'Université Pablo de Olavide (Séville, Espagne).

L'une des motivations principales est alors l'utilisation des cercles et des sphères pour définir des hiérarchies d'occulteurs. En effet, la possibilité de calculer l'orientation des volumes englobants avec des droites permettrait directement de filtrer les occulteurs par groupe dans un arbre de visibilité, en versions complète et paresseuse. Une telle manipulation possède deux avantages directs. Le premier est l'accélération de la construction d'un arbre : reposant principalement sur l'orientation et le filtrage des occulteurs, en travaillant avec une hiérarchie d'occulteurs elle sera automatiquement accélérée. Le second avantage est un gain en terme d'occupation mémoire : puisque nous filtrons une hiérarchie, les données manipulées à un instant t ou encore stockées dans une feuille non encore évaluée seront moins lourdes. En pratique, il s'agira d'un ensemble de nœuds de la hiérarchie ; lorsque ces nœuds seront des feuilles, le coût mémoire sera le même puisqu'il y aura autant de nœuds qu'il y aurait eu d'occulteurs. Par contre nous espérons logiquement traiter plus de nœuds internes de la hiérarchie que de feuilles, pour ainsi réduire la taille de la liste des occulteurs stockée. Cette étude est destinée à se poursuivre en relation avec Frédéric Mora (XLIM/DMI, Limoges).

Projet à moyen terme. Nous proposons d'appliquer à moyen terme les algèbres géométriques à la géométrie discrète en dimension quelconque. Cette étude est programmée en relation avec Eric Andres et Laurent Fuchs (XLIM/SIC, Poitiers).

Pour commencer, nous observons que le modèle conforme s'étend naturellement à un espace géométrique \mathcal{G}_n ; l'algèbre géométrique sur $\mathbb{R}^{n+1,1}$ permet alors de définir des objets de types k -cercles ou k -sphères [Aveneau 2012]. Notons que si le modèle conforme manipule des objets de l'espace Euclidien, c'est principalement parce que la distance d_2 est utilisée dans le plongement de \mathcal{G}_n vers $\mathbb{R}^{n+1,1}$. Cette distance nous permet de définir les objets discrets comme les ensembles de points de \mathbb{Z}^n dont la distance à une courbe de \mathcal{G}_n est inférieure à celle utilisée dans la définition de l'objet discret [Andres 1997].

Ces travaux préliminaires vont se poursuivre, en commençant par la définition d'autres objets discrets reposant sur les algèbres géométriques, par exemple les quadriques en utilisant l'algèbre $\mathbb{R}^{(6,3)}$ [Zamora-Esquivel]. De plus, nous comptons étudier l'impact d'autres plongements de \mathcal{G}_n vers le modèle conforme, *i.e.* ne reposant plus sur la distance d_2 mais par exemple d_1 ou d_∞ ; ceci correspond à l'utilisation de définitions différentes d'objets discrets.

De plus, l'une des problématiques de la géométrie discrète concerne la reconnaissance d'objets de \mathbb{Z}^n , par exemple dans une image ou une vidéo. La reconnaissance de droites dans \mathbb{Z}^2 ou de plans dans \mathbb{Z}^3 semble être relativement maîtrisée. Cependant ce n'est pas le cas dans \mathbb{Z}^n en général, ni de la reconnaissance d'autres formes telles que les cercles, hypersphères, ou plus généralement les k -sphères. Partant de l'observation que $\mathbb{R}^{n+1,1}$ permet de représenter les droites et les cercles par des 3-vecteurs, donc dans un même sous-espace vectoriel, alors il nous semble évident de généraliser les méthodes de reconnaissance de droites en dimension 2 à celles des droites ou cercles en dimension n . Enfin, par extension, cette unification des méthodes doit nous permettre aussi de traiter le cas de la reconnaissance des k -cercles dans \mathbb{Z}^n .

Projet à long terme. Enfin, une troisième voie à explorer à plus long terme est la réécriture de l'équation du rendu directement dans les algèbres. En effet, nous avons exploré l'espace des 2-vecteurs de l'algèbre de Grassmann, qui est celui des droites de \mathfrak{G}_n . Cependant, une algèbre est l'espace vectoriel résultant de l'union des espaces vectoriels des 0 à n vecteurs. Ainsi, elle contient à la fois les points géométrique (les 1-vecteurs), les droites, mais aussi par exemple les hyperplans (les $(n - 1)$ -vecteurs).

L'idée est donc d'écrire l'équation de transport dans l'algèbre (de Grassmann ou dans le modèle conforme). La luminance partant d'un point dans une direction donnée est alors une application de $\wedge^1(\mathbb{R}^n) \times \wedge^2(\mathbb{R}^n)$ vers \mathbb{R} (ou encore dans \mathbb{R}^l avec une luminance sur l longueur d'ondes).

L'intégrale sur les directions incidentes de l'ETL devient une intégrale sur les 2-vecteurs. Dans sa forme sur les surfaces, elle correspond aussi à l'intégrale sur des 1-vecteurs (*i.e.* les points de la scène), en utilisant bien sûr le terme géométrique et la visibilité. Avec les algèbres, il est donc toujours possible de passer de l'une à l'autre des formes de l'ETL. L'idée est d'exploiter cette nouvelle écriture, afin de travailler sur des nouveaux algorithmes de rendu en dimension n , mixant points, droites, plans et bien sûr la visibilité.

Une première étude dans \mathfrak{G}_2 nous permettra d'y voir un peu plus clair, à l'image des travaux de [Ramamoorthi 2007]. Depuis un point x donné, la luminance reçue de tous les points visibles de la scène peut être, en pratique, calculée ou échantillonnée en fonction du signal associé aux différents termes de l'ETL. Reposant sur le côté adimensionnel des algèbres géométriques, les techniques que nous envisageons doivent s'étendre naturellement à \mathfrak{G}_3 bien sûr, mais aussi à \mathfrak{G}_4 , *i.e.* \mathfrak{G}_3 plus le temps.

Notons enfin que ce genre de réécriture (avec les traitements associés) ne doit pas se limiter au seul rendu, bien sûr, mais à n'importe quel problème faisant intervenir les deux espaces \mathfrak{G}_n et \mathfrak{D}_n .

6.2.2 Perspectives en rendu

L'informatique graphique est mon principal domaine de recherche, et donc d'application. Le projet décrit dans cette partie est volontairement limité à la visibilité dans le rendu.

Projet à court terme. Dans le cadre de la première année de thèse de Maxime Maria, nous avons travaillé sur une structure accélératrice pour le calcul de la visibilité le long d'un segment, autrement dit la recherche de la plus proche surface intersectée par un rayon. Cette structure est une partition de l'espace directement déduite de la phase de modélisation topologique d'un bâtiment. Les premiers résultats de ces travaux développés en collaboration avec S. Horna (du laboratoire XLIM/SIC) sont très encourageants, et d'ailleurs actuellement en cours de soumission.

Paradoxalement, notre méthode souffre de deux défauts résultant de ce qui fait sa force : elle repose sur un modèleur topologique, et risque donc d'être d'usage limité à notre laboratoire. De plus, l'ajout de gros ou petit mobilier et autres décorations impose d'utiliser localement une structure annexe. La suite des travaux de thèse de M. Maria sera de corriger ces deux lacunes.

L'idée est la génération d'une structure aux qualités équivalentes mais pour un modèle de bâtiment issu d'un modèleur classique, *i.e.* sans structure topologique. Notre structure actuelle est efficace pour trois raisons principales. D'abord elle est basée sur le modèle de bâtiment, ce qui signifie qu'elle "colle" aux faces telles que les sols, les murs et les plafonds. Ensuite, elle exploite l'ensemble des liaisons topologiques existantes entre sommets, arêtes, faces et volumes. Enfin, la dernière raison est que notre structure est un graphe, et donc que la recherche d'une intersection consiste en un déplacement dans un graphe, sans effectuer d'aller-retour comme pour un BVH ou un kd-tree par exemple. Notons que c'est un principe partagé avec, par exemple, les grilles régulières ; mais notre structure est plus efficace aussi car elle s'adapte à la géométrie pour créer de grands volumes vides (comme le font les BVH et kd-tree).

Dans un premier temps, nous allons construire le graphe sur des bâtiments connus uniquement par leur géométrie. L'idée est relativement simple : nous allons découper l'espace par les plans supports des polygones afin d'obtenir des volumes fermés (les ouvertures étant closes par des portails, ou faces transparentes). En remplaçant les polygones utilisés par nos nouveaux volumes, le graphe associé à la scène sera construit pas à pas.

Pour que le parcours de la structure soit efficace, il faudra néanmoins vérifier que les volumes construits soient de tailles correctes. L'utilisation systématique (au moins dans un premier prototype) de l'heuristique aire-surface (SAH en anglais) permettra de satisfaire cette contrainte de taille. Afin de justifier ce choix, nous nous appuyons sur les travaux sur les constructions d'arbre BSP effectuées dans [Ize 2008].

Par la suite, soit dans la fin de thèse de Maxime Maria, soit dans la prolongation directe de ses travaux, nous allons appliquer ce même principe aux objets quelconques avec l'idée de les inclure directement dans la structure, et donc d'éviter durant le rendu de passer d'une structure accélératrice à une autre, pour encore gagner du temps de calcul.

En résumé, notre objectif dans l'immédiat est de rendre notre structure parfaitement générique et utilisable par quiconque, tout en conservant ce qui fait ses qualités.

Projet à moyen terme. Si la structure développée dans le cadre de la thèse de Maxime Maria est plus efficace qu'un kd-tree ou un BIH², elle est aussi plus économique en mémoire. Ces structures antérieures sont, rappelons-le, de type arbre binaire. La construction d'un kd-tree occupe un espace mémoire difficilement quantifiable à l'avance : lors de la création d'un noeud interne, une partie de la géométrie descend dans la feuille de gauche, une autre dans la feuille de droite, et enfin le reste descend dans les deux feuilles. Cette dernière partie de la géométrie doit donc être dupliquée, impliquant une consommation mémoire. Enfin, le nombre de noeuds d'un kd-tree ou d'un BVH n'est pas prédictible à l'avance, et donc engendre aussi une surconsommation mémoire non décidable. C'est la principale raison expliquant pourquoi ce genre de structures ne peut pas être calculée facilement sur un GPU.

Nous remarquons que ce problème est partagé par nos arbres de visibilité. Or, avec Frédéric Mora nous avons développé un prototype de structure pour le calcul et la représentation de la visibilité. Sa qualité première réside dans sa taille, qui est connue à l'avance : plus précisément, pour n faces bloquantes, elle contient exactement n noeuds. Cette structure peut être utilisée pour les calculs de visibilité, mais peut aussi s'adapter à beaucoup d'autres problèmes. Ainsi, nous l'avons testée dans le cadre du rendu sur GPU de modèles complexes : notre prototype arrive à rendre le modèle "*Thaie Statue*" (10 millions de triangles, disponible sur *The Stanford Graphics Library*) à 30 images par secondes sur une NVidia Titan. Cette nouvelle structure est calculée sur CPU dans notre prototype actuel, mais l'étant en place (un noeud par triangle) nous envisageons ce calcul directement sur GPU (c'est équivalent à un simple tri).

Cette structure est pour l'instant sous forme embryonnaire. Nous devons donc continuer à l'étudier pour vérifier ses qualités, et surtout définir son domaine d'application. En effet, si notre prototype permet de visualiser de gros modèles sur GPU, nous sommes persuadés que cette structure fonctionnera aussi pour d'autres domaines d'application. La première idée est, évidemment, le calcul d'occultation pour faire du rendu dynamique avec ombres douces sur GPU. En effet, nos arbres d'occultations actuels sont trop gros pour être utilisables sur GPU, sauf avec des scènes modestes (quelques dizaines de milliers de triangles). Avec n noeuds pour n occulteurs, la taille de notre nouvelle structure est réduite de façon très importante. Nous pensons donc pouvoir effectuer un rendu avec ombres douces avec des modèles de quelques centaines de milliers de triangles, y compris en environnements dynamiques. Enfin, d'autres utilisations sont à étudier, concernant toutes les applications nécessitant la recherche de convexes dans un espace de dimension donné, sans se limiter au domaine du rendu.

Nous prévoyons au minimum cinq années de recherche pour terminer le développement de cette structure et de ses applications.

Projet à long terme. Notre projet à moyen terme est axé sur une nouvelle structure possédant beaucoup d'applications potentielles. L'une d'elles concerne, évidemment, le calcul de visibilité ou d'occultation. Une application immédiate concerne le rendu sur GPU, en

2. Pour *Bounding Interval Hierarchy*, des arbres binaires mélangeant les concepts de BVH et de kd-tree [Wächter 2006].

raison de la maîtrise de la taille de la structure. Néanmoins, nous avons aussi dans l'idée de réduire la complexité de la construction ou de l'utilisation de nos arbres de visibilité.

Nous avons à long terme le projet de travailler sur la visibilité hiérarchique. L'idée n'est pas ici d'utiliser une hiérarchisation des occulteurs (comme proposé dans les aspects formels au §6.2.1), mais d'utiliser des occulteurs définis par niveaux de détails (ou LOD). Nous avons vu avec nos travaux sur les lobes de cosinus que l'utilisation d'une approximation de la géométrie des occulteurs par un ensemble de sphères permet d'obtenir des ombres de qualité. Le rendu avec ombres douces peut être considéré comme la convolution d'un signal d'occultation par celui de l'émission. Utiliser un modèle simplifié de l'occultation résulte bien souvent en un résultat indiscernable de celui obtenu par le modèle exact.

Cette approche par niveaux de détails est aussi utilisable pour nos autres applications de la visibilité, telles que le calcul de l'occlusion ambiante ou celui de l'illumination indirecte.

Toujours à long terme, nous avons aussi l'ambition de travailler sur de nouvelles applications de la visibilité. Parmi celles-ci, l'une nous apparaît comme particulièrement intéressante. Une approche d'illumination globale bien connue estime l'illumination indirecte en utilisant un ensemble de sources virtuelles, appelées *Virtual Point Lights* (VPL). Ces sources secondaires sont générées par une étape de lancé de rayons de type Monte-Carlo, à l'image de la construction d'une carte de photons. Pour nous, le défaut principal d'une telle méthode est l'utilisation de sources ponctuelles. Or, nous avons montré dans nos travaux précédents que le rendu avec ombres douces est particulièrement rapide tout en restant exact concernant la visibilité. Notre idée est donc d'utiliser des sources surfaciques, pour calculer l'illumination indirecte avec notre méthode de rendu d'ombres douces. Le principal avantage se trouve dans l'aspect doux de l'éclairage direct d'une source surfacique, autorisant à utiliser moins de sources virtuelles qu'avec la méthode VPL classique. En conclusion, nous espérons obtenir plus rapidement des rendus de meilleure qualité.

La difficulté n'est pas dans l'utilisation de sources virtuelles surfaciques, mais dans leurs calculs (position, taille et luminance émise). En réalité, il semble y avoir plusieurs solutions à ce problème, comme par exemple utiliser des sources aléatoires, ou encore effectuer un lancé de faisceaux dans la première phase, *etc.*

Concernant les moyens pour réaliser cette partie du projet à long terme, je compte principalement utiliser des résultats ponctuels de stagiaires de master recherche, et collaborer avec mes estimables collègues de XLIM/DMI et XLIM/SIC (F. Mora, D. Meneveaux, S. Horna, M. Ribardière, *etc.*).

6.2.3 Perspectives en simulation de propagation d'ondes

Cette dernière partie du programme de recherche concerne les futurs développements sous RaPSor, aussi bien pour la radio que l'optique, mais aussi à long terme le domaine acoustique.

Projet à court terme. Le simulateur de propagation RaPSor est construit pour permettre l'utilisation de différents modèles physiques, et non pas seulement celui associé à la radio. Justement, la généralisation des LEDs pour l'éclairage domestique laisse entrevoir des capacités de transport d'information à très faible coût en termes d'infrastructure mais aussi de fonctionnement, avec l'introduction du LiFi³. Néanmoins, tout reste à faire en matière de définition des protocoles réseaux permettant une telle utilisation. Avec Anne Julien-Vergonjanne du département C2S2 de l'institut XLIM, à Limoges, et Pierre Combeau du SIC, nous avons démarré depuis quelques années une collaboration autour de la simulation du canal de propagation pour l'optique sans fil.

Notre objectif est donc de fournir au C2S2 des simulations de propagation dans des environnements complexes et réalistes (en terme de propagation) afin de les aider à définir puis à valider leurs propres études sur les protocoles de communication. De telles simulations sont basées sur une version particulière de l'équation de transport lumineux, calculant non pas la puissance reçue en un pixel, mais une courbe de la puissance reçue en un capteur en fonction du temps (la *réponse impulsionnelle* du canal de propagation). Nos premiers résultats sont très encourageants, et utilisent différents modèles de propagation basés sur une simulation de Monte-Carlo [Combeau 2012, Torkestani 2012].

Ces premiers modèles de calcul restent encore à développer en terme de performances. Nous envisageons ainsi différentes pistes à court terme telles que la parallélisation, ou des méthodes de réduction de variance (échantillonnage par importance, importance multiple, etc.). De plus, nous comptons rapidement développer le travail récent sur la méthode de Metropolis (stage 2013 de Master Recherche de Chau NGuyen). La prise en compte du modèle de communication directement au sein de la méthode de simulation devrait nous permettre de rendre l'outil encore plus efficace.

Notons que ces travaux sur l'optique sans fil vont démarrer dans le cadre du doctorat d'Abdeslam Behloul, qui démarre fin Octobre 2013. L'objectif est la simulation efficace du canal optique pour des systèmes MIMO, *i.e.* comportant n antennes à l'émission et m récepteurs. L'étude utilisera fort logiquement des méthodes bi-directionnelles.

Projet à moyen terme. Une seconde perspective qu'il nous semble important d'étudier concerne les simulations optiques (ou autres) en environnement dynamique, *i.e.* sur une plage de temps assez longue et en incluant la mobilité (des personnes, des émetteurs ou récepteurs, voire du petit ou gros mobilier). L'objectif est de calculer un ensemble discret de simulations statiques pour un pas temporel donné, et ce toujours afin de permettre à nos collègues (du C2S2 ou autre) de tester leurs modèles de communication sur des réponses impulsionnelles réalistes.

Il s'agit d'un environnement idéal pour appliquer nos travaux antérieurs sur la visibilité en dimension n , en utilisant ici un espace de dimension 4 formé de l'espace Euclidien \mathcal{E}_3 auquel nous ajoutons la composante temporelle. Pour accélérer les simulations (en évitant de lancer un calcul complet par pas temporel), nous pensons soit pré-calculer la visibilité

3. *Light Fidelity*, à l'image du protocole de réseaux WiFi, acronyme de *Wireless Fidelity*.

dans \mathcal{D}_4 , soit utiliser une version paresseuse adaptée. Au delà de la méthode de calcul, cette information de visibilité nous semble utilisable en suivant deux pistes prometteuses : dans la première, partant d'une simulation statique dans \mathcal{G}_3 , nous comptons isoler des périodes pour lesquelles certaines contributions ne sont pas impactées par l'animation pour en déduire des (parties de) résultats sur les instants suivants. Dans la seconde, nous tenterons d'explorer directement un calcul dans \mathcal{G}_4 , nécessitant de reformuler l'expression des transferts lumineux dans ce type d'espace, en prenant donc en compte la mobilité. Cette dernière piste est très ambitieuse, mais elle doit trouver des applications non seulement en simulation (optique, radio ou autre), mais aussi en rendu animé (par exemple pour l'industrie du cinéma).

Projet à long terme. L'origine de nos travaux sur la visibilité remonte, historiquement, à nos recherches sur l'optimisation du temps de calcul d'une simulation radio. Les méthodes de calculs de visibilité sont appliquées jusqu'à maintenant principalement au domaine de l'informatique graphique. Certes, l'étude de la visibilité depuis des faces ou des arêtes fût menée dans le cadre d'un contrat France Télécom R&D, mais la livraison contenait une première version qui est depuis longtemps dépassée en terme de performances. Il semble donc assez logique à long terme d'appliquer nos dernières méthodes de calcul de visibilité au domaine de la simulation d'ondes radio. Ceci est d'autant plus évident qu'une des particularités des environnements de simulation dans le domaine de la radio réside dans leur simplicité. En effet, inclure trop de détails géométriques dans les scènes résulte en une sorte de bruit qui rend plus difficile la comparaison d'une simulation avec des mesures dans un environnement réel.

Le projet d'inclure la visibilité dans la simulation est basée sur l'idée d'un graphe de visibilité : pour un noeud donné (une face ou une arête), nous ajoutons un arc vers tous les éléments visibles. Les arbres de visibilité peuvent être conservés, mais l'idée est plutôt de reconstruire la visibilité depuis un point d'émission ou de réception, comme dans notre optimisation basée sur la géométrie. En pratique, l'inconvénient majeur de cette étude réside dans les aspects pratiques : RaPSor est écrit en Java, alors que nos développements en rendu graphique sont effectués en C/C++. Probablement que des projets ou stages d'étudiants en licence et master nous permettront d'avancer pas à pas sur ce sujet.

Un autre aspect qui me semble important concerne la valorisation de notre simulateur RaPSor avec la simulation du son, par exemple en relation avec nos voisins Poitevins de l'institut P'. En effet, les deux physiques actuellement implantées sont la radio et l'optique. Effectuer des simulations sonores est à notre portée, il manque seulement la définition de la physique associée. Cela nécessite principalement le codage des fonctions de réflectance *ad hoc*, et d'outils pour "écouter" le résultat d'une simulation. Couplé à une simulation optique depuis une caméra, utiliser notre simulateur RaPSor pour à la fois se déplacer dans un environnement réaliste avec de l'illumination globale, par exemple avec des VSL (*Virtual Surface Light*), tout en écoutant les sons se réfléchir sur les surfaces telles que le marbre ou le ciment, permettrait par exemple d'effectuer des "ballades" interactives dans des projets de bâtiments architecturaux.

Equation de transfert lumineux

Cette annexe est extraite de la thèse de Romuald Perrot ([Perrot 2012] chapitre 2). Elle présente les principales notions de radiométrie, et l'équation de transfert lumineux (ou ETL) qui décrit les échanges lumineux dans une scène.

A.1 Radiométrie

Les grandeurs essentielles devant être calculées en synthèse d'images appartiennent au domaine visible. Ce dernier est un cas particulier de l'ensemble des rayonnements électromagnétiques. La science étudiant ces rayonnements s'intitule la *radiométrie*.

A.1.1 Angle solide

La notion d'angle solide est très importante pour la simulation d'éclairage. Afin de donner une intuition de sa définition, nous partons de la notion d'angle planaire en dimension 2 et nous étendons celle-ci en dimension 3.

Un angle planaire est défini comme la longueur de l'arc S_p obtenu par projection d'une surface S sur un cercle de centre x et de rayon r (cf. figure A.1). Ainsi, un angle planaire θ est :

$$\theta = \int_{S_p} \frac{dy}{r}.$$

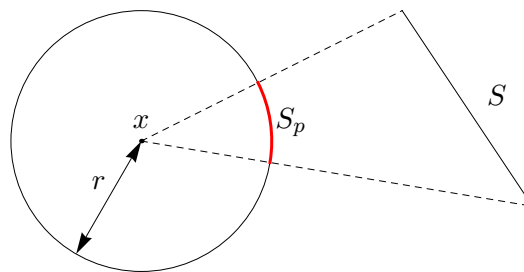


FIGURE A.1 – L'angle planaire sous tendu par une surface S depuis x est proportionnel à la longueur de la projection S_p de S sur le cercle de centre x et rayon r .

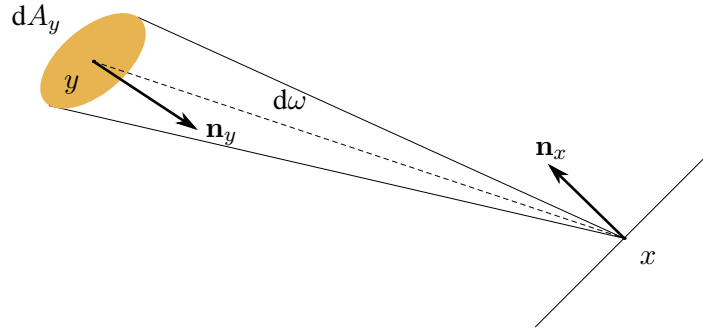


FIGURE A.2 – Angle solide $d\omega$ en fonction d'un élément de surface dA_y .

Cette notion s'étend directement en dimension supérieure et notamment en dimension 3. Soient A_p la projection de la surface A sur la sphère S de rayon r centrée en x ; nous posons :

$$\omega = \int_{A_p} \frac{dA_p}{r^2}.$$

L'élément ω est appelé angle solide centré en x . Son unité est le stéradian, et correspond à l'aire de la surface projetée A_p sur la sphère unité.

Lien entre élément d'angle solide et élément de surface

Les sources lumineuses sont généralement définies en fonction d'une surface et l'intégrale du transfert énergétique est, dans sa forme initiale, exprimée selon des angles solides. La relation suivante permet de faire le lien entre un angle solide élémentaire et une surface, et donc d'exprimer l'ensemble des calculs de la simulation d'éclairage sous une forme unifiée :

$$d\omega_{xy} \approx \frac{(\mathbf{n}_y \cdot \mathbf{yx}) dA_y}{\|x - y\|^2} \quad (\text{A.1})$$

où $d\omega_{xy}$ est l'angle solide centré en x autour d'une direction xy (cf. figure A.2).

A.1.2 Grandeurs radiométriques

A.1.2.1 Energie

L'énergie transportée par un photon pour une longueur d'onde donnée est notée e_λ est :

$$e_\lambda = \frac{hc}{\lambda},$$

où :

- h est la constante de Planck, $h = 6,626\,068\,96 \times 10^{-34} J.s$;
- c est la vitesse de la lumière dans le vide, $c = 299\,792\,458 m.s^{-1}$;

- λ est la longueur d'onde considérée, en mètre.

L'énergie d'un photon s'exprime en Joule (J). L'énergie spectrale Q_λ représente l'énergie d'un ensemble de photons pour une longueur d'onde λ donnée :

$$Q_\lambda = n_\lambda e_\lambda = n_\lambda \frac{hc}{\lambda},$$

où n_λ est le nombre de photons considérés. L'unité est encore le Joule (J).

Les unités précédentes ne sont définies que pour des longueurs d'onde particulières ; l'énergie Q quantifie l'énergie spectrale mais pour l'ensemble des longueurs d'onde :

$$Q = \int_0^\infty Q_\lambda d\lambda.$$

L'unité est toujours le Joule (J).

A.1.2.2 Flux

L'énergie totale d'un ensemble de photons pour toutes les longueurs d'onde pendant un temps t est le flux Φ :

$$\Phi = \frac{dQ}{dt}$$

L'unité est le Watt (W), ou le Joule par seconde ($J.s^{-1}$).

C'est par cette grandeur que les sources lumineuses sont définies en synthèse d'images. Nous dirons par exemple qu'une source lumineuse émet 100 W : peu importe la forme de la source et la distribution de l'émission de l'énergie, s'agissant ici du flux total émis par cette source.

A.1.2.3 Radiosité - Eclairément

A partir du flux, deux autres notions importantes sont établies : la radiosité et l'éclairément. Elles définissent, respectivement, le flux émis et le flux reçu par unité de surface. La radiosité d'une unité de surface centrée en x est notée $B(x)$ et l'éclairément est noté $E(x)$. Leurs définitions sont les suivantes :

$$B(x) = \frac{d\Phi_o}{dA_x}$$

$$E(x) = \frac{d\Phi_i}{dA_x}$$

où Φ_i et Φ_o sont les flux respectivement reçus et émis en x . L'unité est le Watt par mètre carré ($W.m^{-2}$). La figure A.3 illustre ces deux notions.

Lorsque le contexte lève toute ambiguïté, le flux (incident ou sortant) sera noté Φ .

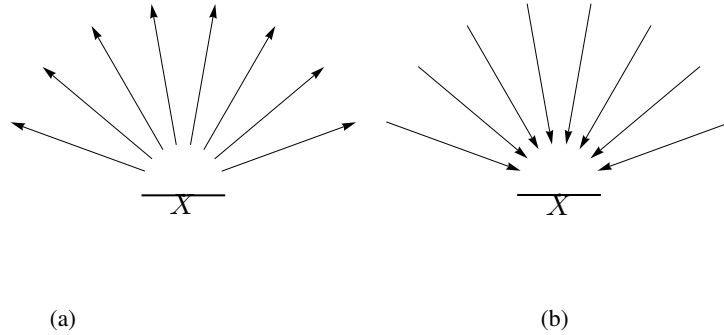


FIGURE A.3 – Radiosité et éclairage. (a) La radiosité caractérise le flux émis par une surface. (b) L'éclairage caractérise le flux reçu par la surface.

A.1.2.4 Intensité

Pour un angle solide $\omega_{\mathbf{v}}$, l'intensité rayonnante $I(\mathbf{v})$ est définie de la façon suivante :

$$I(\mathbf{v}) = \frac{d\Phi}{d\omega_{\mathbf{v}}}.$$

L'unité est le Watt par stéradian ($W \cdot sr^{-1}$). La figure A.4 donne une illustration d'intensité sortante.

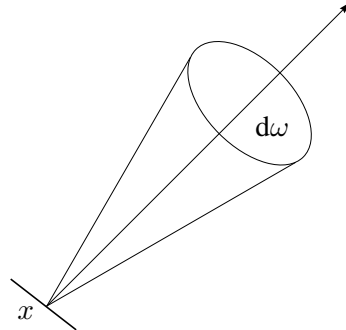


FIGURE A.4 – Intensité lumineuse en un point. Cette grandeur est la quantité d'énergie par unité d'angle solide $d\omega$.

A.1.2.5 Luminance

La dernière notion de radiométrie que nous présentons est la luminance. Elle est essentielle à la simulation d'éclairage : l'oeil humain y est sensible et non pas à l'intensité ni à l'éclairage. La luminance intègre à la fois une notion de surface et de direction. Sa définition est la suivante :

$$L(x, \mathbf{v}) = \frac{d^2\Phi}{\cos \theta dA_x d\omega_{\mathbf{v}}}$$

où θ est l'angle entre la normale à la surface centrée en x et la direction \mathbf{v} . Souvent cette relation est donnée à partir du produit scalaire :

$$L(x, \mathbf{v}) = \frac{d^2\Phi}{(\mathbf{n}_x \cdot \mathbf{v}) dA_x d\omega_{\mathbf{v}}}.$$

La luminance est le terme le plus important en synthèse d'images puisque c'est elle qui indique l'aspect final d'un objet, et qui permet de rendre compte des effets tels que la brillance ou l'anisotropie. Notons que si la luminance peut être définie par longueur d'onde, nous considérons cette dernière comme muette afin de simplifier le discours.

A.1.2.6 Relations entre grandeurs

Parfois il est nécessaire d'avoir des mécanismes pour passer d'une quantité à une autre. Ci-dessous sont listées quelques relations usuelles permettant de lier la luminance à chacune des autres quantités :

$$\Phi = \int_A \int_{\Omega} L(x, \mathbf{v}) \cos \theta d\omega_{\mathbf{v}} dA_x, \quad (\text{A.2})$$

$$E(x) = \int_{\Omega} L_i(x, \mathbf{v}) \cos \theta d\omega_{\mathbf{v}}, \quad (\text{A.3})$$

$$I(\mathbf{v}) = \int_A L(x, \mathbf{v}) dA_x. \quad (\text{A.4})$$

A.1.3 Propriétés

Cette section décrit deux relations sur lesquelles les algorithmes de simulation reposent.

A.1.3.1 Luminance constante le long d'une droite

Cette propriété est fondamentale dans tous les algorithmes de simulation d'éclairage. Elle stipule que dans le vide, la luminance émise à un point x dans la direction d'un point y est égale à la luminance reçue par y depuis le point x :

$$L(x, \mathbf{xy}) = L(y, \mathbf{yx}).$$

En voici une démonstration considérant des surfaces différentielles :

$$\begin{aligned} L(x, \mathbf{xy}) &= \frac{d^2\Phi}{(\mathbf{n}_x \cdot \mathbf{xy}) dA_x d\omega_{\mathbf{xy}}} \\ &= \frac{d^2\Phi}{(\mathbf{n}_x \cdot \mathbf{xy}) dA_x \frac{(\mathbf{n}_y \cdot \mathbf{yx}) dA_y}{\|x-y\|^2}} \\ &= \frac{d^2\Phi}{(\mathbf{n}_y \cdot \mathbf{yx}) dA_y d\omega_{\mathbf{yx}}} \\ &= L(y, \mathbf{yx}). \end{aligned}$$

A.1.3.2 Flux d'un émetteur diffus

Cette propriété lie très fortement la luminance et le flux d'une source lumineuse surfacique diffuse, *i.e.* une source lumineuse omnidirectionnelle émettant de la lumière uniformément dans toutes les directions. Dire qu'une source lumineuse est diffuse signifie que sa luminance $L(x, \mathbf{o})$ est constante quel que soit le point x et quelle que soit la direction \mathbf{o} . Ceci a pour conséquence directe la relation suivante :

$$\begin{aligned}
 \Phi &= \int_A \int_{\Omega} L(x, \mathbf{v}) \cos \theta \, d\omega_{\mathbf{v}} \, dA_x \\
 &= \int_A \int_{\Omega} L \cos \theta \, d\omega_{\mathbf{v}} \, dA_x \\
 &= L \int_A \int_{\Omega} \cos \theta \, d\omega_{\mathbf{v}} \, dA_x \\
 &= LA \int_{\Omega} \cos \theta \, d\omega_{\mathbf{v}} \\
 &= LA\pi.
 \end{aligned}$$

Les sources lumineuses surfaciques sont souvent considérées comme des émetteurs diffus. Dans ce cas, cette formule facilite le lien entre le flux, la luminance et l'aire de la surface de la source.

A.2 Fonction de distribution de réflectance bidirectionnelle

A.2.1 Présentation

Lorsqu'un rayon lumineux atteint une surface, une partie du flux est dispersée tandis qu'une autre peut être absorbée ou encore transmise (cas des surfaces transparentes ou translucides par exemple). La manière dont la lumière est dispersée au contact de la surface est modélisée par une *fonction de distribution de dispersion bidirectionnelle* (FDDB). Celle-ci peut être décomposée en deux fonctions complémentaires, une fonction de distribution de réflectance bidirectionnelle (FDRB) et une fonction de distribution de transmittance bidirectionnelle (FDTB). La première caractérise le phénomène de réflexion d'une surface, *i.e.* la manière dont les rayons lumineux sont réfléchis dans le même milieu d'incidence. La seconde est spécifique aux rayons lumineux transmis à travers un dioptré.

La FDRB est le rapport entre la luminance réfléchie dans une direction donnée par un objet et l'éclairement incident :

$$f_r(x, \mathbf{i}, \mathbf{o}) = \frac{dL_o(x, \mathbf{o})}{dE(x, \mathbf{i})} = \frac{dL_o(x, \mathbf{o})}{L_i(x, \mathbf{i})(\mathbf{n}_x \cdot \mathbf{i}) \, d\omega_i}$$

où $f_r(x, \mathbf{i}, \mathbf{o})$ est la FDRB d'un élément de surface centré en x pour une direction incidente \mathbf{i} et une direction sortante \mathbf{o} , $L_i(x, \mathbf{i})$ et $L_o(x, \mathbf{o})$ sont appelées respectivement luminance incidente et sortante.

Note : nous écrivons $f_r(x, \mathbf{i}, \mathbf{o})$ une FDRB, $f_t(x, \mathbf{i}, \mathbf{o})$ une FDTB et $f(x, \mathbf{i}, \mathbf{o})$ une Fddb.

A.2.2 Propriétés

Les propriétés des FDRB permettent d'exprimer de manière formelle des intuitions physiques.

A.2.2.1 Positivité

Les FDRB sont définies positives :

$$f_r(x, \mathbf{i}, \mathbf{o}) \geq 0.$$

Une FDRB ne peut pas être négative puisque par définition elle est le rapport entre la luminance réfléchie et l'éclairement.

A.2.2.2 Conservation de l'énergie

Afin de respecter la loi de conservation de l'énergie, l'intégrale d'une FDRB sur l'hémisphère quelle que soit la direction incidente \mathbf{i} est bornée :

$$\int_{\Omega_x} f_r(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{o}) d\omega_{\mathbf{o}} \leq 1. \quad (\text{A.5})$$

Cette définition est également naturelle : une surface ne peut pas renvoyer plus d'énergie qu'elle n'en reçoit.

Nota bene : l'intégrale de cette inéquation est appelée réflectivité et est notée ρ :

$$\rho = \int_{\Omega_x} f_r(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{o}) d\omega_{\mathbf{o}}. \quad (\text{A.6})$$

A.2.2.3 Réciprocité

Dernière propriété souvent utilisée en optique : la loi de réciprocité, ou critère de Helmholtz ou encore loi de symétrie. Elle stipule qu'intervertir les directions d'incidence et de réflexion dans le calcul de la FDRB à un point x ne change pas sa valeur. Ainsi on a :

$$f_r(x, \mathbf{i}, \mathbf{o}) = f_r(x, \mathbf{o}, \mathbf{i}).$$

Ceci signifie que quel que soit le sens de propagation du flux lumineux, la valeur de la FDRB pour deux directions \mathbf{i} et \mathbf{o} est la même. Cette propriété est utilisée, en particulier, dans de nombreux algorithmes de simulation d'éclairage, reposant soit sur la propagation des flux lumineux soit sur la collecte de flux.

Nota bene : cette propriété n'est pas vraie pour les FDTB (cf. [Veach 1997]).

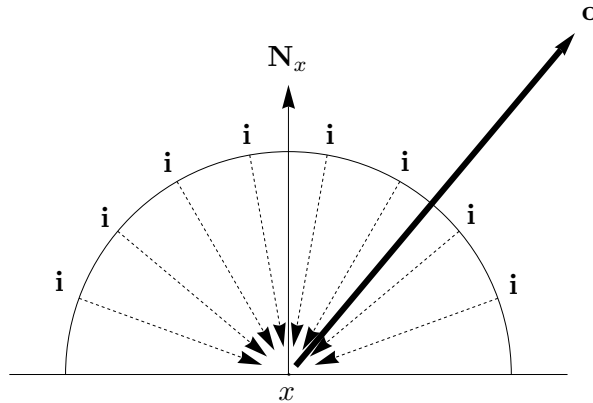


FIGURE A.5 – Termes utilisés dans l’ETL selon l’hémisphère. L’intégrale représente la luminance sortante pour un élément de surface centré en x dans la direction \mathbf{o} , comme la somme des contributions réfléchies provenant des directions \mathbf{i} . L’auto-émission en x est volontairement omise.

A.3 Equation de transfert lumineux

L’équation de transfert lumineux (ETL), aussi appelée équation du rendu dans un contexte de synthèse d’images, exprime le flux se propageant dans la scène à partir des sources lumineuses. Cette équation peut s’énoncer de deux manières différentes : une intégrale définie selon les hémisphères et une selon les aires des surfaces. La première calcule le flux lumineux provenant d’un ensemble de directions et la seconde effectue le même calcul mais pour un ensemble de surfaces. Cette deuxième forme est notamment utilisée pour calculer les contributions des sources lumineuses.

A.3.1 Intégration sur l’hémisphère

L’ETL permet de calculer la luminance émise depuis un point x dans une direction \mathbf{o} . Sa formulation, donnée par Kajiya [Kajiya 1986], est la suivante :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_{\Omega_x} L_i(x, \mathbf{i}) f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i. \quad (\text{A.7})$$

La luminance émise depuis un élément de surface centré en x dans une direction \mathbf{o} est donc la luminance intrinsèque L_e à laquelle s’ajoute la somme des luminances reçues et ré-émises dans la direction \mathbf{o} . La figure A.5 illustre les différents termes utilisés dans l’ETL.

Cette équation prend uniquement en compte les phénomènes de réflexion. Trois modifications de l’ETL permettent d’inclure les phénomènes de transmission :

- l’intégration sur la sphère \mathcal{S}_x centrée en x (et non plus sur l’hémisphère) ;

- l'utilisation d'une fonction de distribution de dispersion bidirectionnelle, combinant à la fois une fonction de distribution de réflectance bidirectionnelle et une fonction de distribution de transmission bidirectionnelle ;
- l'ajout d'une valeur absolue autour du calcul du cosinus entre la normale en x et la direction \mathbf{i} .

L'ensemble de ces trois notions permet d'écrire une équation plus générale :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_{S_x} L_i(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o}) |\mathbf{n}_x \cdot \mathbf{i}| d\omega_{\mathbf{i}}. \quad (\text{A.8})$$

A.3.2 Intégration sur les aires

Dans l'intégrale (A.7), le calcul de la luminance incidente provient d'un ensemble de directions. Il est parfois plus pratique de manipuler une intégration utilisant les luminances émises en chaque point de la scène.

La transformation de l'équation (A.8) est effectuée à partir de l'équation (A.1) :

$$\begin{aligned} L_o(x, \mathbf{o}) &= L_e(x, \mathbf{o}) + \int_{\Omega_x} L_i(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o}) |\mathbf{n}_x \cdot \mathbf{i}| d\omega_{\mathbf{i}} \\ &= L_e(x, \mathbf{o}) + \int_A L_i(x, \mathbf{xy}) f(x, \mathbf{xy}, \mathbf{o}) \frac{|\mathbf{n}_x \cdot \mathbf{xy}| |\mathbf{n}_y \cdot \mathbf{yx}|}{\|x - y\|^2} V(x, y) dA_y, \end{aligned}$$

ce qui conduit à l'intégrale sur les aires suivantes :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_A L_i(x, \mathbf{xy}) f(x, \mathbf{xy}, \mathbf{o}) G(x, y) V(x, y) dA_y \quad (\text{A.9})$$

où $G(x, y) = \frac{|\mathbf{n}_x \cdot \mathbf{xy}| |\mathbf{n}_y \cdot \mathbf{yx}|}{\|x - y\|^2}$ est appelé *terme géométrique*, A représente l'ensemble des surfaces de la scène et $V(x, y)$ est un terme de visibilité indiquant si x et y sont mutuellement visibles.

A.3.3 Discussion sur l'équation de transfert lumineux

L'ETL peut être réécrite sans perte de généralité suivant plusieurs termes, chacun étant associé à une sémantique différente. Ce partitionnement est couramment mis en œuvre afin de traiter chacun des termes par une méthode adaptée. Une séparation des termes de l'équation est effectuée en distinguant, dans la partie réflexion de l'équation, la luminance L_i^L reçue depuis les sources lumineuses, de celle L_i^S reçue depuis les autres surfaces :

$$L_i(x, \mathbf{i}) = L_i^L(x, \mathbf{i}) + L_i^S(x, \mathbf{i}).$$

Ainsi, l'équation A.7 devient :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_{\Omega_x} L_i(x, \mathbf{i}) f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i \quad (\text{A.10})$$

$$= L_e(x, \mathbf{o}) + \int_{\Omega_x} [L_i^L(x, \mathbf{i}) + L_i^S(x, \mathbf{i})] f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i \quad (\text{A.11})$$

$$= L_e(x, \mathbf{o}) + \int_{\Omega_x} L_i^L(x, \mathbf{i}) f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i \quad (\text{A.12})$$

$$+ \int_{\Omega_x} L_i^S(x, \mathbf{i}) f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i$$

$$= L_{\text{emis}} + L_{\text{direct}} + L_{\text{indirect}}. \quad (\text{A.13})$$

De cette façon, l'équation de transfert lumineux est ré-écrite en trois termes :

- L_{emis} , un terme émissif intrinsèque à la surface au point x , sa valeur est non nulle pour une source lumineuse ;
- L_{direct} , un terme direct représentant la contribution des sources lumineuses ;
- L_{indirect} , un terme indirect représentant la contribution des autres surfaces, ou, autrement dit, celle des sources lumineuses mais après une ou plusieurs réflexions sur les surfaces de la scène.

Ce partitionnement permet d'adapter le mode de calcul de chacun des termes. Par exemple, le terme d'éclairage direct repose souvent sur l'ETL exprimée selon les surfaces.

Méthodes de Monte-Carlo et tracé de chemins

Cette annexe partiellement extraite du chapitre 2 de la thèse de Romuald Perrot [Perrot 2012] présente les méthodes de Monte-Carlo, et leur application à l'ETL avec le suivi de chemins.

B.1 Intégration de Monte-Carlo

Les méthodes de Monte-Carlo permettent d'approcher numériquement une intégrale sur un domaine de dimension quelconque [Hammersley 1967].

B.1.1 Rappels de probabilité

Nous rappelons ici les notions de probabilité nécessaires à l'introduction des méthodes de Monte-Carlo.

B.1.1.1 Notion de probabilité

Un *événement aléatoire* X est un événement ayant une certaine chance d'apparaître. La *probabilité* $P(X)$ est la mesure de cette chance, et est telle que $0 \leq P(X) \leq 1$.

La notation $P(X_1 + X_2 + \dots + X_N)$ est la mesure de la chance d'apparition d'au moins un des événements X_1, X_2, \dots, X_N . La notation $P(X_1 X_2 \dots X_N)$ est la mesure de la chance d'apparition de tous les événements X_1, X_2, \dots, X_N . Rappelons les deux axiomes suivants :

$$\begin{aligned} P(X_1 + X_2 + \dots + X_N) &\leq P(X_1) + P(X_2) + \dots + P(X_N), \\ P(X_1 X_2 \dots X_N) &= P(X_1) P(X_2) \dots P(X_N). \end{aligned}$$

Lorsque $P(X_1 + X_2) = P(X_1) + P(X_2)$, alors X_1 et X_2 sont dit événements *exclusifs*. Lorsque $P(X_1 + X_2 + \dots + X_N) = 1$, alors les événements X_1, X_2, \dots, X_N sont dits *exhaustifs*.

La notation $P(A|B)$ est la *probabilité conditionnelle*, *i.e.* la chance que A apparaisse sachant B . Si $P(A|B) = P(A)$, alors A et B sont dits *indépendants*.

B.1.1.2 Variable aléatoire, distribution et espérance

Soit un ensemble d'événements aléatoires à la fois exhaustifs et exclusifs, caractérisé par un nombre η . Ce nombre η est appelé *variable aléatoire*. Un exemple trivial est la variable aléatoire associant le numéro de la face supérieure obtenu après un lancé d'un dé à 6 faces, *i.e.* un nombre dans $\{1, 2, 3, 4, 5, 6\}$.

La *fonction de distribution*¹ F est définie comme la probabilité $F(y) = P(\eta \leq y)$. Par exemple, pour un dé non pipé, nous avons $F(\cdot \leq 0) = 0$, $F(1) = \frac{1}{6}$, $F(2) = \frac{2}{6}$, ..., $F(\cdot \geq 6) = 1$. Cette fonction de distribution admet les propriétés évidentes suivantes : $F(-\infty) = 0$, $F(+\infty) = 1$, et $x \leq y \Rightarrow F(x) \leq F(y)$.

Soit la fonction $g(\eta)$. L'*espérance mathématique* $E[g(\eta)]$ est la moyenne pondérée par leur probabilité des événements $g(\eta)$:

$$E[g(\eta)] = \int g(y) dF(y).$$

Lorsque F admet une dérivée $f(y)$, alors nous avons :

$$E[g(\eta)] = \int g(y)f(y) dy.$$

Si F est une fonction discrète f_i au point y_i , alors :

$$E[g(\eta)] = \sum_i g(y_i) f_i.$$

Les termes $f(y)$ et f_i sont les *fonctions fréquentielles* de la variable aléatoire η . Le premier est aussi connu comme la *fonction de densité de probabilité* (ou *pdf*). Un exemple discret est encore pour le dé : l'espérance d'un tirage d'un dé non pipé à 6 faces est alors $E[\eta] = \frac{1}{6} \sum_{i=1}^6 i = \frac{21}{6} = 3,5$.

B.1.1.3 Variables aléatoires indépendantes

Soit l'ensemble d'événements à la fois exhaustifs et exclusifs, caractérisé par les deux variables aléatoires η et ξ et de fonction de distribution $F(\eta, \xi)$. Nous pouvons construire les deux nouveaux ensembles d'événements suivants :

$$F(y, z) = P(\eta \leq y, \xi \leq z), \quad G(y) = P(\eta \leq y), \quad H(z) = P(\xi \leq z).$$

Les variables aléatoires η et ξ sont dites *indépendantes* si :

$$F(y, z) = G(y)H(z) \text{ pour tous } y \text{ et } z.$$

La relation suivante est toujours vraie :

$$\sum_i E[g_i(\eta_i)] = E \left[\sum_i g_i(\eta_i) \right].$$

1. Ou encore *fonction de distribution cumulative*.

Pour un ensemble de variables aléatoires η_i indépendantes, la relation suivante est vraie :

$$\prod_i E [g_i(\eta_i)] = E \left[\prod_i g_i(\eta_i) \right].$$

B.1.1.4 Moment d'ordre k

Par définition, le n ème *moment* de η est $E[\eta^n]$; pour $n = 1$, alors $\mu = E[\eta]$ est appelée la *moyenne* de η .

Par définition, $\mu_n = E[(\eta - \mu)^n]$ est le n ème *moment central* de η ; pour $n = 2$, alors $\mu_2 = E[(\eta - E[\eta])^2]$ est dit la *variance* de η .

Les termes $\sigma = \sqrt{\mu_2}$ et σ/μ sont respectivement appelés l'*écart-type* et le *coefficient de variation*.

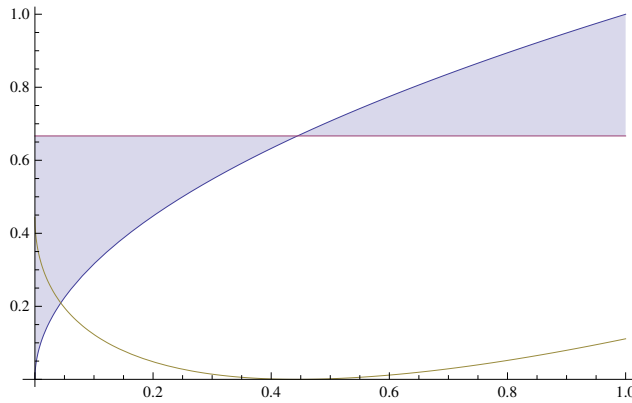


FIGURE B.1 – Fonction $g(y) = \sqrt{y}$ (en bleu), sa moyenne $\mu = 2/3$ (en rouge) et $(\sqrt{y} - \mu)^2$, la dérivée de sa variance (en jaune verdâtre).

Prenons un exemple illustré sur la figure B.1 : soit la distribution $F(y) = y$ pour $y \in [0 \dots 1]$, et soit la fonction $g(y) = \sqrt{y}$. La distribution est correctement normalisée puisque :

$$\int_0^1 dF(y) = 1.$$

De plus, F est continue et sa densité de probabilité est constante ($f(y) = 1$). La moyenne de y est :

$$\mu = \int_0^1 \sqrt{y} f(y) dy = \left[\frac{2}{3} y^{\frac{3}{2}} \right]_0^1 = \frac{2}{3},$$

et sa variance est :

$$\mu_2 = \int_0^1 (\sqrt{y} - \mu)^2 dy = 0.05555556.$$

L'écart-type est donc $\sigma = 0.235702$: il mesure la dispersion moyenne de la fonction \sqrt{y} par rapport à sa moyenne μ .

B.1.1.5 Principe de l'estimation

L'estimation consiste à estimer la valeur inconnue d'un paramètre θ quelconque d'une distribution donnée. Le paramètre θ est appelé l'*estimande*. Le processus part toujours d'un *échantillon* η , consistant en une suite d'observations (η_1, \dots, η_n) dont la taille n est dite *taille de l'échantillon*. Lorsque l'observation est associée à une variable aléatoire X , l'estimateur noté $\hat{\theta}(X)$ est aussi une variable aléatoire.

L'estimande θ est donc un paramètre de la distribution des variables aléatoires constituant l'échantillon η . Ce paramètre peut être par exemple la variance ou l'espérance mathématique. Pour estimer θ , la fonction $\hat{\theta}(\eta)$ possède les sémantiques suivantes :

1. Soit l'*estimateur* $\hat{\theta}(\mathbf{y})$: pour des variables quelconques $y_1 \dots y_n$.
2. Soit l'*estimé* $\hat{\theta}(\theta)$: appliqué sur les observations η .

Le problème est donc d'obtenir un bon estimateur fournissant des estimations correctes de θ , *i.e.* de choisir $\hat{\theta}(\mathbf{y})$ tel que $\hat{\theta}(\eta)$ soit proche de θ . Puisque η est aussi une distribution, alors $\hat{\theta}(\eta)$ l'est aussi ; sa *distribution d'échantillonnage* est :

$$T(u) = P(\hat{\theta}(\eta) \leq u) = \int_{\hat{\theta}(\mathbf{y}) \leq u} dF(\mathbf{y})$$

Nous définissons le *biais* β et la variance d'échantillonnage σ_t^2 de $\hat{\theta}$ comme :

$$\begin{aligned} \beta &= E \left[\hat{\theta}(\eta) - \theta \right] \\ &= \int (\hat{\theta}(\mathbf{y}) - \theta) d\mathbf{y} \\ \sigma_t^2 &= E \left[\left(\hat{\theta}(\eta) - E \left[\hat{\theta}(\eta) \right] \right)^2 \right] \\ &= E \left[\left(\hat{\theta} - \theta - \beta \right)^2 \right] \\ &= \int \left(\hat{\theta}(\mathbf{y}) - \theta - \beta \right)^2 dF(\mathbf{y}) \end{aligned}$$

Lorsque β et σ_t^2 sont petits, nous disons que $\hat{\theta}(\mathbf{y})$ est un *bon estimateur*. Plus précisément :

- Si $\beta = 0$, nous parlons d'un *estimateur non biaisé*.
- Si $\sigma_t^2 \leq \sigma^2$, nous parlons d'un *estimateur à variance minimale*.

Un estimateur très utile en pratique est le suivant :

- η_1, \dots, η_n sont des observations *indépendantes* d'une même distribution parente.
- Un estimateur non-biaisé de la moyenne de cette distribution parente est :

$$\hat{\eta} = \frac{\eta_1 + \eta_2 + \dots + \eta_n}{n}.$$

Il a pour erreur standard (σ étant la déviation standard de la distribution parente) :

$$\sigma_{\hat{\eta}} = \frac{\sigma}{\sqrt{n}}.$$

– Un estimateur non biaisé de la variance de la distribution parente est :

$$s^2 = \frac{\eta_1^2 + \eta_2^2 + \dots + \eta_n^2 - n\hat{\eta}^2}{n-1},$$

dont l'erreur standard est :

$$\sigma_{s^2} \approx \frac{\sigma^2}{\sqrt{n/2}}.$$

B.1.2 Principe de base

L'estimation de Monte-Carlo vise à calculer l'intégrale d'une fonction f sur un domaine Ω de dimension m donné :

$$I = \int_{\Omega} f(\mathbf{x}) d\mathbf{x}.$$

Pour $m = 1$ le calcul peut être effectué avec une méthode numérique classique (Simpson, Romberg, Gauss ...). L'inconvénient principal de ces dernières réside dans leur erreur de calcul, qui est en $\sigma/n^{1/m}$. Si pour une petite dimension m cela reste raisonnable, lorsque $m \gg 1$ le coût du calcul rend rapidement les méthodes impraticables.

Reposant sur la loi des grands nombres, une estimation de Monte-Carlo approche le calcul de l'intégrale I :

$$\hat{I} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)}$$

où p est une *pdf* uniforme, respectant donc les deux propriétés suivantes :

$$\forall \mathbf{x} \in \Omega, p(\mathbf{x}) \geq 0, \tag{B.1}$$

$$\int_{\Omega} p(\mathbf{x}) d\mathbf{x} = 1. \tag{B.2}$$

B.1.3 Exemple : calculer π

Prenons un exemple simple, sur un domaine de dimension 1 : le calcul de la valeur de π . Cette méthode de calcul de π est loin d'être la meilleure, elle ne sert que d'exemple de l'estimation de Monte-Carlo.

L'aire du disque de rayon r est πr^2 . Il s'agit de l'intérieur d'un cercle $x^2 + y^2 = r^2$. Il est donc possible de calculer π comme l'aire d'un disque de rayon unitaire, que l'on approche par l'intégrale suivante :

$$\frac{\pi}{4} = \int_0^1 \sqrt{1-x^2} dx.$$

La figure B.2 illustre la fonction intégrée (en bleu), sa moyenne (en rouge) et sa déviation (en jaune).

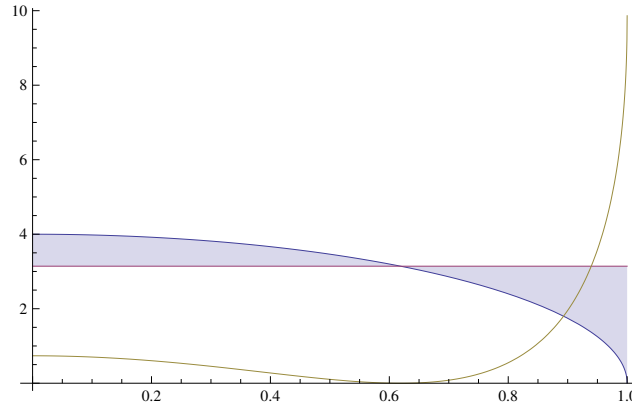


FIGURE B.2 – Exemple d’estimation de la fonction $4\sqrt{1-x^2}$ sur $[0 \dots 1]$ (en bleu), de moyenne $\mu = \pi$ (en rouge), et de variance $\sigma^2 = \frac{32}{3} - \pi^2 = 0.892783$ (sa fonction dérivée est en jaune verdâtre).

Prenons un échantillon de taille 16 ; nous avons :

#	1	2	3	4	5	6	7	8
X_i	0.7446	0.3057	0.0715	0.1865	0.8750	0.9584	0.5573	0.7484
$f(X_i)$	2.6699	3.8085	3.9898	3.9298	1.9365	1.1415	3.3213	2.6531
$(f(X_i) - \mu)^2$	0.2225	0.4448	0.7194	0.6213	1.4522	4.0002	0.0323	0.2386

#	9	10	11	12	13	14	15	16
X_i	0.8146	0.7262	0.8718	0.6505	0.2746	0.7223	0.0440	0.2610
$f(X_i)$	2.3199	2.7500	1.9594	3.0379	3.8463	2.7665	3.9961	3.8614
$(f(X_i) - \mu)^2$	0.6751	0.1534	1.3976	0.0108	0.4966	0.1407	0.7302	0.5181

Nous obtenons alors les résultats suivants :

- Estimateur de π : $\hat{\pi} = 2.99925$.
- La variance théorique est : $\sigma_\pi^2 = 0.740854$.
- L’écart-type théorique est : $\sigma_\pi = 0.860729$.
- La variance de la distribution parente : $\mu_2 = \sum_{i=1}^{16} \frac{(f(X_i)^2 - 16\hat{\pi}^2)}{15} = 0.768631$.
- L’erreur standard est : $\sigma_{\hat{\pi}}/\sqrt{n/2} = \sqrt{\sigma_{\hat{\pi}}^2/8} = 0.309966$.

L’erreur est importante. Nous avons vu qu’elle dépend de la taille de l’échantillon.

En prenant 64 observations nous avons :

- Estimateur de π : $\hat{\pi} = 3.15386$.
- La variance théorique : $\sigma_\pi^2 = 0.743138$.
- L’écart-type théorique : $\sigma_\pi = 0.862055$.
- La variance parente : $\mu_2 = \sigma_{\hat{\pi}}^2 = 0.754781$.
- L’erreur standard : $\sigma_{\hat{\pi}}/\sqrt{n/2} = 0.15358$.

Avec $n = 10^6$ observations il vient :

- Estimateur de π : $\hat{\pi} = 3.14168$.

- Variance théorique : $\sigma_{\pi}^2 = 0.796097$.
- L'écart-type théorique : $\sigma_{\pi} = 0.892243$.
- La variance parente : $\mu_2 = \sigma_{\pi}^2 = 0.796098$.
- L'erreur standard : $\sigma_{\hat{\pi}}/\sqrt{n/2} = 0.000892243$.

En pratique, l'estimateur définit un intervalle $[\hat{\theta} - 2\sigma, \hat{\theta} + 2\sigma]$ sur lequel la *bonne valeur* réside à 95 %. L'erreur standard est donc le paramètre le plus important à étudier lors d'une estimation : la seule valeur de l'estimateur est insuffisante, c'est bien l'intervalle de confiance qui nous renseigne vraiment.

B.1.4 La recherche de la vérité

L'exemple précédent du calcul de π montre que la méthode semble fonctionner, bien qu'elle nécessite un grand nombre d'observations pour conduire à une bonne estimation. Nous exprimons plus formellement ici la correction de la méthode, en calculer l'espérance de notre estimateur de Monte-Carlo :

$$\begin{aligned}
 E[\hat{I}] &= \int_{\Omega} \hat{I} p(\mathbf{x}) d\mathbf{x} \\
 &= \int_{\Omega} \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} p(\mathbf{x}) d\mathbf{x} \\
 &= \int_{\Omega} f(\mathbf{x}) d\mathbf{x} \\
 &= I
 \end{aligned}$$

Si le biais est nul, ce n'est par contre pas le cas de la variance :

$$\begin{aligned}
 V[\hat{I}] &= E \left[\left(\hat{I} - E[\hat{I}] \right)^2 \right] \\
 &= E[\hat{I}^2] - E[\hat{I}]^2 \\
 &= \int_{\Omega} \hat{I}^2 p(\mathbf{x}) d\mathbf{x} - I^2 \\
 &\leq \frac{\sigma^2}{N} \text{ (théorème central limite)}
 \end{aligned}$$

Ainsi, il faut 4 fois plus d'observations pour diviser par 2 l'erreur standard.

B.1.5 Amélioration de la convergence

Sans entrer dans les détails, nous mentionnons néanmoins l'existence de méthodes de réduction d'erreur.

La méthode *stratifiée* consiste à diviser le domaine d'intégration en strates, et à effectuer une estimation dans chacun d'eux. En choisissant correctement les strates et la taille de l'échantillon dans chacun d'entre elles, alors la variance est presque à coup sûr largement réduite.

Une autre méthode améliorant la convergence est l'*échantillonnage d'importance*. Elle consiste à utiliser une distribution non uniforme des observations. L'idéal est alors de transformer la fonction à intégrer en une densité de probabilité, et à l'utiliser pour générer un échantillon de taille unitaire. Alors, l'estimateur est à coup sûr l'estimande. Le seul problème est la transformation de la fonction à intégrer en densité, qui passe par une normalisation par son intégrale sur le domaine, ce qui est absurde. En pratique, nous utilisons des fonctions de densité "proches" de la fonction à intégrer, mais ayant une intégrale connue.

Une troisième méthode porte le nom de *variables de contrôle*. Elle consiste à utiliser une fonction auxiliaire à la fois facile à intégrer sur le domaine, et proche de la fonction à intégrer. En effet, en écrivant :

$$I = \int_{\Omega} g(\mathbf{x}) d\mathbf{x} + \int_{\Omega} [f(\mathbf{x}) - g(\mathbf{x})] d\mathbf{x},$$

l'estimateur est alors :

$$\hat{I} = \int_{\Omega} g(\mathbf{x}) d\mathbf{x} + \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\mathbf{x}_i) - g(\mathbf{x}_i)}{p(\mathbf{x}_i)}.$$

L'erreur est donc réduite, considérant que f et g sont proches.

B.1.6 Roulette russe

La roulette russe est une méthode permettant d'utiliser moins d'échantillons dans un estimateur tout en donnant un résultat similaire (*i.e.* l'espérance du nouvel estimateur est la même que celle de l'ancien). Le principe général est de ne garder qu'une portion des échantillons pour évaluer l'estimateur. Soit p une probabilité, et ξ_i un nombre aléatoire uniformément distribué ; alors la contribution $f_{rr}(x_i)$ d'un échantillon x_i est la suivante :

$$f_{rr}(x_i) = \begin{cases} \frac{f(x_i)}{p} & \text{si } \xi_i < p, \\ 0 & \text{sinon.} \end{cases}$$

En d'autres termes, seuls les échantillons pour lesquels $\xi_i < p$ sont utilisés dans l'estimateur. Ce mécanisme donne un nouvel estimateur non biaisé lorsque l'estimateur de départ ne l'est pas non plus. En effet, l'espérance d'un estimateur basé sur une roulette russe est :

$$E[\hat{I}_{rr}] = (1 - p) \times 0 + p \times \frac{E[\hat{I}]}{p} = E[\hat{I}].$$

Cette méthode est souvent utilisée en synthèse d'images pour arrêter des processus récursifs infinis tout en assurant un résultat non biaisé.

B.2 Tracé de chemins

Le tracé de chemins [Kajiya 1986] est, historiquement, la méthode de référence pour la production d'images de qualité permettant l'utilisation de tous types de surfaces (implicites, explicites, *etc*), de tous types de FDRB (diffuses, brillantes, transparentes, *etc*) et de sources lumineuses (ponctuelles, surfaciques, *etc*). En effet, sa construction reposant directement sur l'ETL et sur les méthodes de Monte-Carlo, est consistante et non biaisée.

B.2.1 Méthode de base

Le terme émissif de l'ETL est directement calculable, la difficulté réside dans l'intégrale du second terme. Pour approcher celle-ci, Kajiya propose d'abord de remplacer le calcul de la luminance incidente par la luminance sortante dans la direction $-\mathbf{i}$; celle-ci est directement issue du point $y = \tau(x, \mathbf{i})$, *i.e.* le point d'intersection entre le rayon (x, \mathbf{i}) et la scène :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_{S_x} L_o(\tau(x, \mathbf{i}), -\mathbf{i}) f(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_{\mathbf{i}}.$$

La deuxième étape consiste à appliquer une méthode de Monte-Carlo afin de calculer une approximation de l'intégrale :

$$\hat{L}_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \frac{1}{N} \sum_{k=1}^N \frac{L_o(\tau(x, \mathbf{i}_k), -\mathbf{i}_k) f(x, \mathbf{i}_k, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}_k)}{p(\mathbf{i}_k)},$$

où \mathbf{i}_k est la direction choisie après la $k^{\text{ème}}$ réflexion et $p(\mathbf{i}_k)$ est la probabilité d'avoir tiré cette direction.

Cette définition est cependant toujours récursive, et appliquée directement elle conduit à une explosion combinatoire. En effet, elle signifie qu'à chaque point d'intersection $y = \tau(x, \mathbf{i})$, N nouveaux rayons sont générés. Pour résoudre ce problème, Kajiya propose de ne construire qu'un seul chemin (en fixant $N = 1$), *i.e.* un seul rayon est relancé à chaque réflexion. Ainsi, le processus consiste à construire de manière incrémentale (*cf.* figure B.3) un chemin dans la scène puis à l'évaluer.

La valeur numérique calculée est alors la suivante :

$$\hat{L}_o(x, \mathbf{o}) = \sum_{i=1}^N \mathcal{P}_i(x, \mathbf{o})$$

où $\mathcal{P}_i(x, \mathbf{o})$ est la contribution du chemin i . Une formulation possible de cette contribution est la suivante :

$$\mathcal{P}_i(x, \mathbf{o}) = \sum_{k=0}^{\infty} \mathcal{P}^k(x_k, \mathbf{i}_{k-1})$$

où $\mathcal{P}^k(x_k, \mathbf{i}_{k-1})$ est la contribution du chemin après k interactions avec la scène. Cette expression de la contribution d'un chemin indique que l'on ajoute les contributions au fur

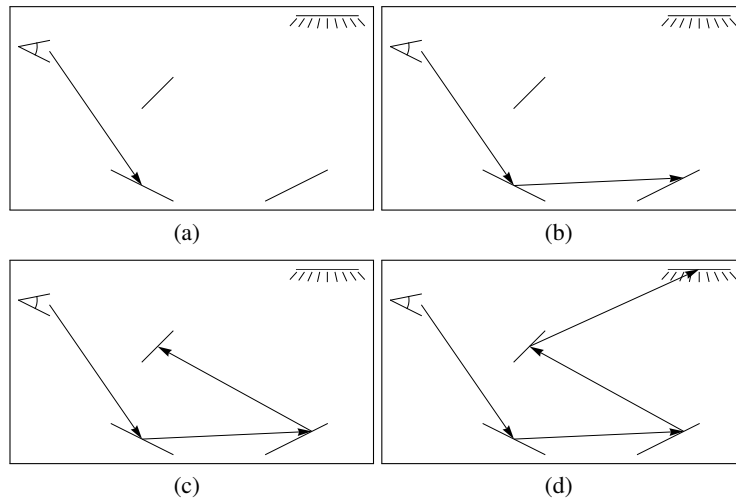


FIGURE B.3 – Construction incrémentale des chemins dans une méthode de tracé de chemins. À chaque surface atteinte, une nouvelle direction est choisie.

et à mesure que le chemin est construit. Ainsi :

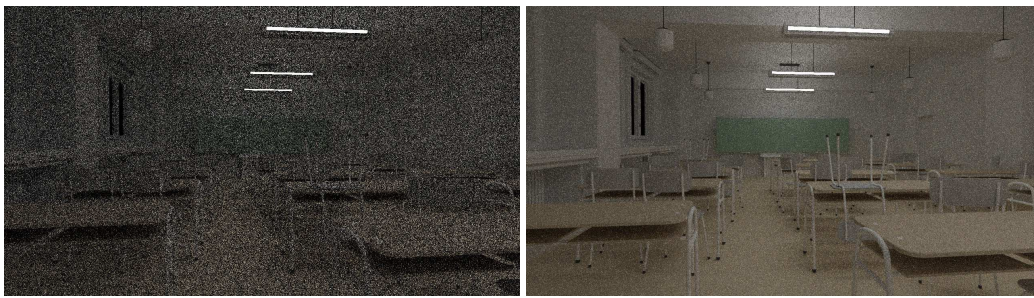
$$\begin{aligned}
 \mathcal{P}^0(x_0, \mathbf{i}_{-1}) &= L_e(x, \mathbf{o}) \\
 \mathcal{P}^1(x_1, \mathbf{i}_0) &= f_r(x, \mathbf{o}, \mathbf{i}_0)(\mathbf{n}_x \cdot \mathbf{i}_0)L_e(x_1, -\mathbf{i}_0) \\
 \mathcal{P}^2(x_2, \mathbf{i}_1) &= f_r(x, \mathbf{o}, \mathbf{i}_0)(\mathbf{n}_x \cdot \mathbf{i}_0)f_r(x, -\mathbf{i}_0, \mathbf{i}_1)(\mathbf{n}_{x_1} \cdot \mathbf{i}_1)L_e(x_2, -\mathbf{i}_1) \\
 \mathcal{P}^3(x_3, \mathbf{i}_2) &= \dots
 \end{aligned}$$

La longueur du chemin est cependant toujours finie ; si une borne sur la longueur des chemins était ajoutée alors l'estimateur serait biaisé. Une roulette russe est mise en œuvre pour limiter la longueur des chemins sans imposer de borne maximale. Ainsi, un test est effectué après chaque surface atteinte pour décider si le chemin doit être poursuivi.

Le principal problème du tracé de chemins est sa vitesse de convergence. Afin d'obtenir une erreur faible, il est nécessaire d'explorer l'ensemble de la scène (matériaux, objets, sources lumineuses). Pour certaines scènes complexes (FDRB, sources lumineuses et géométries complexes), ceci requiert la construction de plusieurs dizaines de milliers de chemins. Or ce coût de construction est important à cause de l'évaluation de la fonction lancée de rayons. Par conséquent, pour certaines configurations, les temps de calculs peuvent atteindre plusieurs dizaines d'heures. La figure B.4 donne des exemples de rendus obtenus par la méthode de tracé de chemins pour différentes tailles d'échantillon.

B.2.2 Calcul avec estimation du prochain événement

Le chemin a une contribution non nulle seulement si un de ses noeuds est situé sur une source lumineuse. Lorsqu'une scène comporte des sources lumineuses ponctuelles ou de très petites tailles la probabilité pour que cet événement survienne est très faible. Dans un tel cas, beaucoup de contributions sont nulles et donc la variance est très forte.



(a) 1 chemin par pixel.

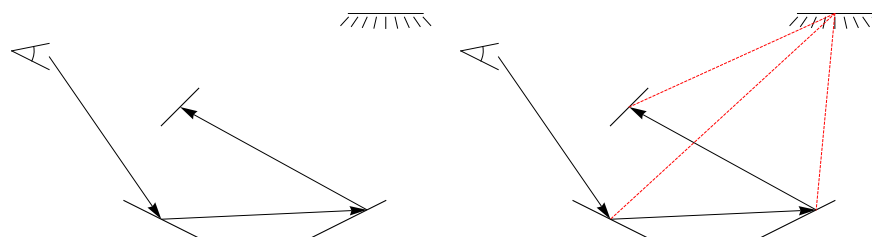
(b) 32 chemins par pixel.



(c) 128 chemins par pixel.

(d) 2048 chemins par pixel.

FIGURE B.4 – Rendu par tracé de chemins.

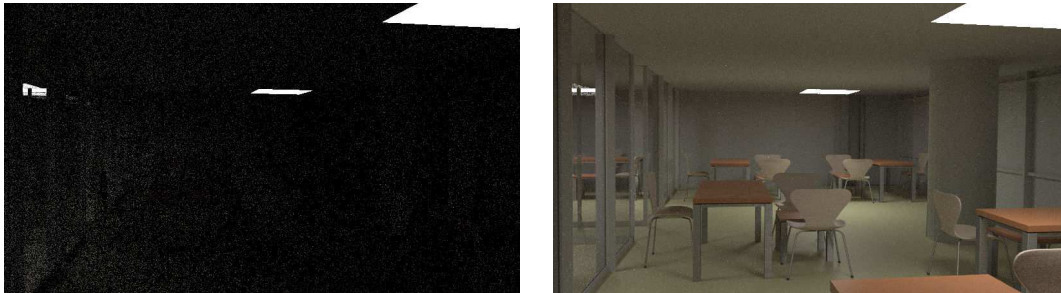


(a) Tracé de chemins classique

(b) Avec estimation du prochain événement

FIGURE B.5 – Tracé de chemins sans et avec calcul des contributions directes.

Dutré [Dutre 1996] suggère d'utiliser une méthode appelée *estimation du prochain événement*. L'idée est d'appliquer une version partitionnée de l'ETL (direct et indirect, cf. section A.3.3) ; ainsi, à chaque réflexion une contribution directe est évaluée. Ceci permet une réduction de la variance puisque la probabilité d'obtenir une contribution non nulle par chemin augmente. La figure B.5 montre le processus effectué lors de l'évaluation d'un chemin. La figure B.6 montre une comparaison entre un simple tracé de chemins et la version avec prise en compte des contributions directes.



(a) Sans prise en compte des contributions directes.

(b) Avec estimation du prochain événement.

FIGURE B.6 – Tracé de chemins pour un même temps de calcul.

Estimation de densité et carte de photons

Cette annexe est extraite du chapitre 2 de la thèse de Romuald Perrot [Perrot 2012]. Elle présente les méthodes d'estimation de densité, suivies de la méthode des cartes de photons.

C.1 Estimation de densité

C.1.1 Présentation générale

Les méthodes d'estimation de densité permettent de calculer une approximation d'une fonction. En effet, très souvent, cette fonction n'est pas connue. Par exemple, dans un contexte expérimental, seul un échantillon de petite taille est disponible. L'estimation permet alors de construire une approximation de la fonction à partir de cet échantillon.

Les méthodes d'estimation de densité peuvent avoir un *a priori* sur la nature de la densité observée. Par exemple, si la fonction de densité est de nature normale (*i.e.* gaussienne), le problème revient alors à chercher la moyenne et l'écart-type de cette distribution. Il s'agit, dans ce cas, de méthodes d'*estimation paramétrique*. Un exemple classique de ce type de méthode est l'estimation par méthode des moindres carrés. En revanche, les méthodes *non-paramétriques* ne supposent aucune connaissance de la fonction ou de la nature de l'échantillon. La suite de cette annexe se place dans le cadre des méthodes non-paramétriques.

C.1.2 Estimateur naïf

C.1.2.1 Principe

Nous avons vu dans le §B.1.1.2 de l'annexe B le lien existant entre une fonction de distribution F et la densité f :

$$\frac{dF(x)}{dx} = f(x).$$

En appliquant la définition de la dérivée, le calcul de la fonction de densité de probabilité devient :

$$f(x) = \lim_{h \rightarrow 0} \frac{F(x+h) - F(x-h)}{2h}.$$

En utilisant l'estimateur $\hat{F}(x)$ de $F(x)$ suivant :

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n \chi_{\{\alpha < x\}}(x_i),$$

où (x_i) est un échantillon de taille n , et $\chi_A(\alpha)$ est la fonction caractéristique indiquant l'appartenance d'une valeur donnée à un ensemble A :

$$\chi_A(\alpha) = \begin{cases} 1 & \text{si } \alpha \in A, \\ 0 & \text{sinon,} \end{cases}$$

alors l'estimateur naïf $\hat{f}(x)$ de f est :

$$\hat{f}(x) = \frac{\hat{F}(x+h) - \hat{F}(x-h)}{2h}, \quad (\text{C.1})$$

$$= \frac{1}{2hn} \sum_{i=1}^n \chi_{\{x-h \leq \alpha < x+h\}}(x_i). \quad (\text{C.2})$$

Dans le contexte de l'estimation de densité, le terme h est appelé *fenêtre*.

C.1.2.2 Exemple numérique

Nous donnons un exemple numérique d'estimation de densité de probabilité. Soit la fonction f_{1u} définie sur l'intervalle $[-5; 10]$ suivante :

$$f_{1u}(x) = 0.020x^5 - 0.0258x^4 - 0.0142x^3 + 0.8002x^2 - 0.1873x + 0.8331.$$

Elle est normalisée de façon à respecter l'équation B.2 :

$$f_1(x) = \frac{f_{1u}(x)}{\int_{-5}^{10} f_{1u}(v) dv}.$$

La figure C.1 donne l'allure de la courbe $f_1(x)$ et la reconstruction par estimation de densité en utilisant un échantillon de taille 500.

C.1.2.3 Variation de la fenêtre

La fenêtre est un paramètre très sensible. En effet, elle indique le degré de lissage. Une fenêtre de très faible taille entraîne une sensibilité importante aux variations de l'échantillonnage. Une fenêtre large permet de lisser l'estimation. Le choix de la fenêtre idéale est très compliqué. Celle-ci peut être définie en fonction des considérations de l'estimateur global (désir d'une faible variance, d'un faible biais), mais cette approche masque des effets locaux (une fenêtre large va lisser des pics sans doute désirés). La figure C.2 donne différentes courbes obtenues à partir d'un même échantillon (*cf.* figure C.1(b)), seule la taille de la fenêtre diffère.

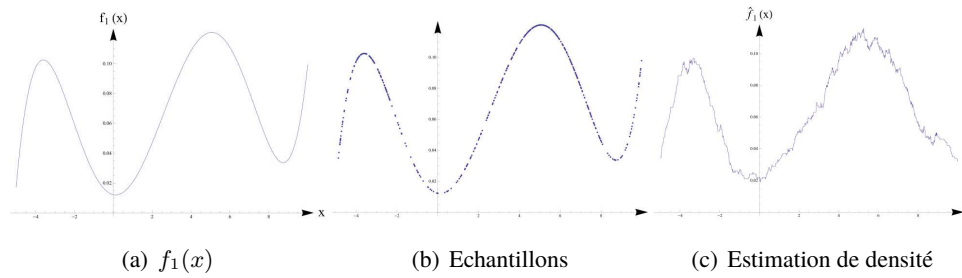


FIGURE C.1 – Exemple d’estimation de densité de la fonction f_1 avec un estimateur naïf et une fenêtre $h = 1$. (a) Fonction exacte. (b) Échantillon. (c) Estimation de densité.

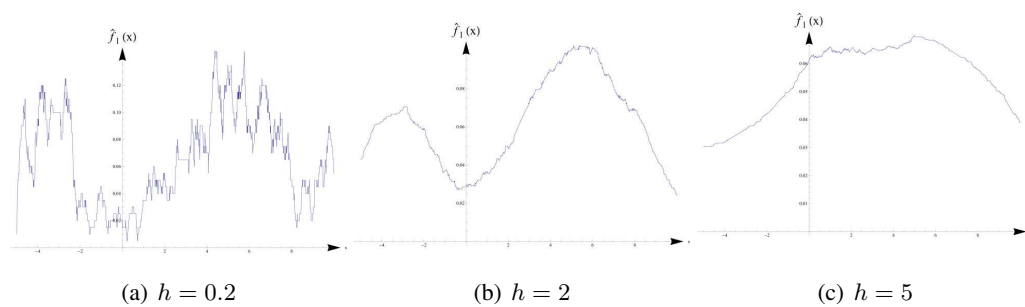


FIGURE C.2 – Exemple d’estimation de densité en faisant varier la taille de la fenêtre. L’échantillon sur lequel est effectué l’estimation de densité est le même. Plus la fenêtre augmente et plus les variations de l’estimation sont lissées.

Des méthodes adaptent automatiquement la taille de la fenêtre en fonction de la densité des échantillons. Lorsque peu de valeurs de l’échantillon x_i sont présents autour du point d’estimation x , le paramètre h est grand, tandis que lorsque la densité d’échantillonnage est importante autour de x alors h est petit. La valeur exacte de h est donnée par la distance $d_k(x)$ du $k^{\text{ème}}$ plus proche voisin x_k de x . La méthode s’appelle alors *estimation de densité par k plus proches voisins*.

C.1.3 Estimateur par noyaux

C.1.3.1 Principe

L’estimateur naïf peut être généralisé dans une classe d’estimateurs appelés *estimateurs par noyaux* :

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K_1\left(\frac{x - x_i}{h}\right),$$

où K_1 est appelée noyau. L’estimateur naïf peut être exprimé à l’aide de la *fenêtre de Parzen* [Parzen 1962] :

$$K_1(x) = \begin{cases} \frac{1}{2} & \text{si } |x| < 1 \\ 0 & \text{sinon.} \end{cases}$$

Un noyau est généralement choisi avec les propriétés suivantes :

$$\forall \mathbf{x} \in \mathbb{R}^d, K_d(\mathbf{x}) \geq 0 \quad (\text{positif}), \quad (\text{C.3})$$

$$\int_{-\infty}^{\infty} K_d(\mathbf{x}) \, d\mathbf{x} = 1 \quad (\text{normalisé}), \quad (\text{C.4})$$

$$\int_{-\infty}^{\infty} \mathbf{x} K_d(\mathbf{x}) \, d\mathbf{x} = 0 \quad (\text{symétrique}), \quad (\text{C.5})$$

$$0 \leq \sigma_K^2 = \int_{-\infty}^{\infty} \mathbf{x}^2 K_d(\mathbf{x}) \, d\mathbf{x} < \infty \quad (\text{borné}). \quad (\text{C.6})$$

Les deux équations C.3 et C.4 indiquent que le noyau K_d est aussi une densité de probabilité ; l'équation (C.5) indique que K_d est une fonction symétrique ; enfin, l'équation (C.6) permet d'exprimer le biais comme une fonction de σ_K^2 (cf. le paragraphe suivant). Ces propriétés ne sont pas obligatoires à l'estimation de densité.

Plusieurs fonctions ont été proposées pour K_1 . La figure C.3 en donne une liste non exhaustive extraite de la littérature.

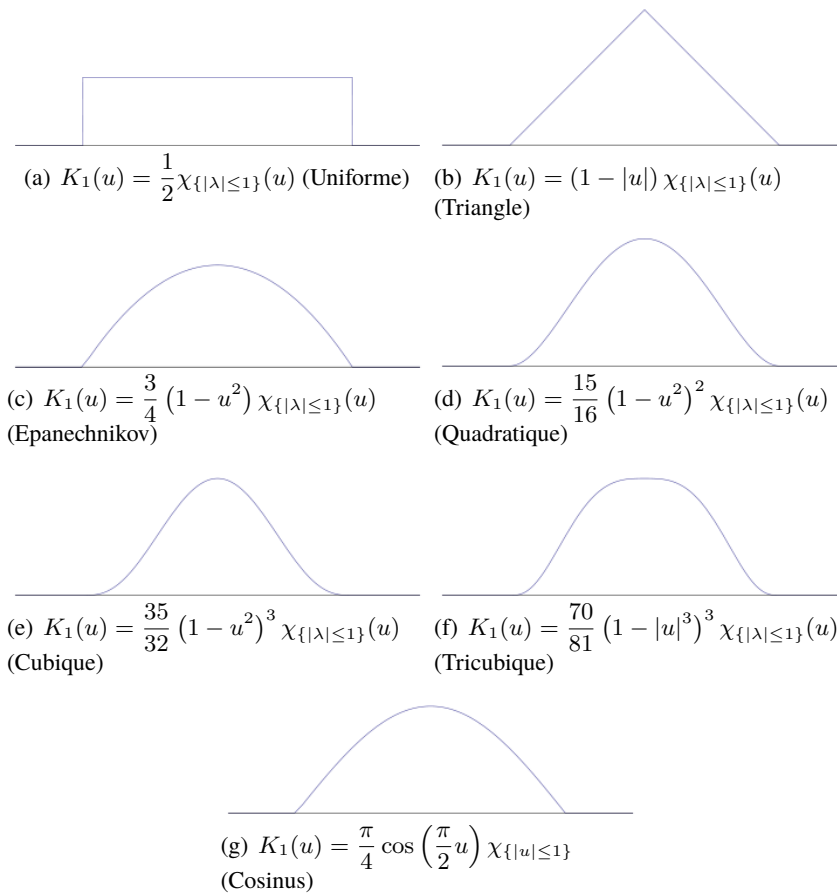


FIGURE C.3 – Exemples de noyaux. La courbe donne la pondération affectée à une valeur de l'échantillon en fonction de sa distance vis-à-vis du point d'estimation.

La principale motivation pour le choix d'un noyau réside dans le problème de l'influence

lointaine. En effet, dans une approche naïve, lorsqu'une valeur d'échantillon est proche ou distante du point d'estimation, sa contribution est constante. Ceci pose problème car une valeur au bord de la fenêtre a une influence identique à ce que peut produire une autre très proche du point d'estimation. Les différents types de noyaux permettent de donner une influence d'une valeur de l'échantillon en fonction de sa distance par rapport au point d'estimation.

C.1.4 Biais dans l'estimation de densité

L'estimation de densité est donc intrinsèquement biaisée. Rappelons la définition du biais :

$$b(\hat{f}_h(\mathbf{x})) = E[\hat{f}(\mathbf{x})] - f(\mathbf{x}),$$

où E est l'espérance mathématique.

En supposant que f soit dérivable deux fois, le biais peut s'écrire pour un noyau K_1 (cf. [Silverman 1998]) :

$$\begin{aligned} b(\hat{f}_h(x)) &= f(x) \int_{-\infty}^{\infty} K_1(z) dz - f(x) - hf'(x) \int_{-\infty}^{\infty} zK_1(z) dz \\ &\quad + \frac{h^2}{2} f''(x) \int_{-\infty}^{\infty} z^2 K_1(z) dz + o(h^2). \end{aligned} \quad (\text{C.7})$$

Clairement, les propriétés C.3 à C.6 trouvent ici leur utilité. L'équation (C.4) permet d'annuler les deux premiers termes. L'équation (C.5) annule le troisième. En utilisant l'équation (C.6), il vient :

$$\begin{aligned} b(\hat{f}_h(x)) &= \frac{h^2}{2} f''(x) \int_{-\infty}^{\infty} z^2 K_1(z) dz + o(h^2), \\ &= \frac{h^2}{2} f''(x) \sigma_K^2 + o(h^2). \end{aligned}$$

En conclusion, le biais est *a priori* proportionnel à h^2 . Avec une petite largeur de bande, ce comportement est généralement considéré comme acceptable.

C.1.5 Estimation en dimension supérieure

Les méthodes d'estimation de densité sont utilisables en toutes dimensions. Une estimation de densité par noyaux sur un espace de dimension d est :

$$\hat{f}(\mathbf{x}) = \frac{1}{n|H|} \sum_{i=1}^n K_d(H^{-1}(\mathbf{x} - \mathbf{x}_i)),$$

où H est une matrice carrée inversible de taille d , $|H|$ son déterminant, H^{-1} son inverse, K_d un noyau de dimension d . Si la matrice est diagonale :

$$H = \begin{pmatrix} h_1 & & 0 \\ & \ddots & \\ 0 & & h_d \end{pmatrix},$$

Alors chaque élément h_i contrôle le lissage sur chacune des dimensions. De la même manière, si $H = hI_d$, où I_d est la matrice identité de dimension d , alors l'estimateur est :

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K_d \left(\frac{\mathbf{x} - \mathbf{x}_i}{h} \right).$$

Le choix du noyau a peu d'influence sur la variance de l'estimateur [Silverman 1998]. Ainsi, le noyau peut être radial symétrique, comme par exemple le noyau d'Epanechnikov en dimension 2 :

$$K_{es}(\mathbf{x}) = \frac{2}{\pi} (1 - \|\mathbf{x}\|^2) \chi_{\{\|\mathbf{x}\|^2 \leq 1\}},$$

ou bien, être directement le produit de d noyaux de dimension 1 :

$$K_d(\mathbf{x}) = \prod_{i=1}^d K_1(x_i),$$

soit dans le cas du noyau de type Epanechnikov en dimension 3 :

$$K_{ep}(\mathbf{x}) = \frac{27}{64} (1 - x^2) (1 - y^2) (1 - z^2),$$

où (x, y, z) sont les coordonnées de \mathbf{x} .

C.2 Cartes de photons

C.2.1 Introduction

La mise en œuvre des cartes de photons repose sur deux étapes : (i) la simulation des inter-réflexions lumineuses et son stockage dans une structure appelée *carte de photons* ; (ii) l'exploitation de ce calcul pour la réalisation de l'image finale.

Les calculs d'inter-réflexions sont réalisés en construisant des chemins issus des sources lumineuses. Chaque impact, appelé *photon*, d'un chemin avec la scène est stocké dans une carte de photons. Celle-ci est ensuite organisée de façon à pouvoir accélérer la recherche des photons voisins de n'importe quel point de la scène. Les avantages de cette structure sont nombreux. Elle est indépendante du point de vue : une même carte de photons pourra être réutilisée pour générer des images depuis des points de vue différents dans le cas où aucun paramètre de la scène ne change. Un autre avantage notable est la construction de cette structure, qui ne prend que quelques secondes. Enfin, reposant sur une approche rigoureuse de type Monte-Carlo, la simulation effectuée est consistante et non biaisée.

La deuxième étape consiste à utiliser les résultats de la simulation d'éclairage pour produire une image. Cette phase, dépendante du point de vue, commence par déterminer les surfaces visibles depuis la caméra ; puis, pour chaque point visible, elle récupère l'ensemble des photons dans un voisinage donné à partir de ceux-ci, et en déduit une valeur de luminance par une méthode d'estimation de densité [Simonoff 1996]. Puisque l'interpolation des valeurs est effectuée avec des photons proches mais n'étant pas situés exactement au point d'estimation, la méthode est intrinsèquement biaisée.

On notera que la carte de photons n'est pas toujours utilisée pour produire l'image finale. Brouillat [Brouillat 2008] propose d'utiliser une carte de photons afin de construire un cache d'éclairage. L'image est ensuite produite à partir du cache. Néanmoins, la construction du cache repose toujours sur les méthodes usuelles d'estimation de densité décrites dans la suite de cette annexe.

C.2.2 Détail de la méthode

Les formulations suivantes sont celles introduites par Veach [Veach 1997] avec les notations de Pharr *et al.* [Pharr 2010]. Le tracé de photons consiste à produire des chemins lumineux depuis les sources. Chaque impact d'un chemin avec une surface de la scène produit un photon. Ce dernier est identifié par une position x_i , une direction incidente \mathbf{i}_i et une énergie α_i où i est l'indice du photon. Dans la terminologie de Veach [Veach 1997], α_i est également appelé *poids* du photon. Son calcul est donné par les relations suivantes¹ :

$$\alpha^0 = \frac{L_e(y, \mathbf{o})(\mathbf{n}_y \cdot \mathbf{o})}{p(y, \mathbf{o})}$$

$$\alpha^{j+1} = \alpha^j \frac{f_r(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{o})}{p(\mathbf{o})}$$

où α^0 est le poids d'une particule émise sur une source lumineuse depuis la position y dans la direction \mathbf{o} , \mathbf{n}_x est la normale à la surface considérée au point x et $p(y, \mathbf{o})$ est la probabilité d'avoir émis la particule depuis la position y dans la direction \mathbf{o} . Le poids α^j est celui de la particule après j rebonds sur les surfaces de la scène. Le dernier rebond s'effectuant au point x depuis la direction incidente \mathbf{i} dans la direction sortante \mathbf{o} avec une probabilité $p(\mathbf{o})$. Enfin, L_e est l'émissivité de la surface.

Veach [Veach 1997] montre que des poids définis de la sorte respectent l'égalité suivante :

$$E \left[\frac{1}{N} \sum_{j=1}^N \alpha_j W_e(x_j, \mathbf{v}_j) \right] = \int_A \int_{S^2} W_e(x, \mathbf{i}) L_i(x, \mathbf{i}) \, d\mathbf{i} \, dA, \quad (\text{C.8})$$

où $W_e(x, \mathbf{i})$ est appelée *fonction de mesure* et $(\alpha_j, x_j, \mathbf{v}_j)$ sont respectivement le poids, la position et la direction incidente d'un photon.

1. Les indices représentent un numéro de photon dans un chemin quelconque, alors que les exposants désignent des photons d'un même chemin.

La deuxième étape consiste, pour un point x (appelé *point d'estimation*) et une direction \mathbf{o} , à calculer une approximation de la luminance indirecte. Veach montre que ce calcul revient à choisir une fonction de mesure adéquate :

$$L_{ind}(x, \mathbf{o}) = \int_S L(x, \mathbf{i}) f_r(x, \mathbf{o}, \mathbf{i}) (\mathbf{n}_x \cdot \mathbf{i}) \, d\mathbf{i}, \quad (\text{C.9})$$

$$= \int_A \int_S W_e(x, \mathbf{i}) L_i(x, \mathbf{i}) \, d\mathbf{i} \, dA, \quad (\text{C.10})$$

avec $W_e(x, \mathbf{i})$:

$$W_e(x, \mathbf{i}) = \delta(x - x_j) f_r(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}). \quad (\text{C.11})$$

En pratique, cette équation est toujours fautive puisque la probabilité d'avoir un photon exactement au point x est quasiment nulle. Dans les cartes de photons, cette contrainte est relâchée :

$$L_{ind}(x, \mathbf{o}) = \sum_{i=1}^k \frac{\alpha_i f_r(x, \mathbf{i}_i, \mathbf{o})}{A} K\left(\frac{x - x_i}{h}\right)$$

où k est le nombre de photons utilisés dans l'estimation de densité, α_i est le poids du $i^{\text{ème}}$ photon, A est l'aire d'un disque de rayon h centré en x , x_i est la position du photon i et $K(u)$ est une fonction d'interpolation appelée *noyau*.

Nota bene : dans sa forme initiale, Jensen [Jensen 2001] utilise une estimation par k plus proches voisins. Néanmoins d'autres auteurs conservent une estimation de densité par noyaux [Schjøth 2007]. Cette deuxième forme permet non seulement de conserver une estimation continue mais également indépendante des données. Nous nous plaçons dans ce cadre, car il simplifie les raisonnements et les démonstrations sans perdre en qualité d'estimation.

C.2.3 Limitations

La méthode souffre de nombreux défauts. Le choix du nombre de photons à stocker et à utiliser lors de l'estimation de densité n'est pas trivial. En effet, ceux-ci sont non seulement dépendants des scènes utilisées mais contrairement au tracé de chemins la simple augmentation d'un paramètre n'indique pas nécessairement une diminution de l'erreur.

Ce biais est usuellement classé en trois catégories [Schregle 2003] :

- Le *biais topologique*, il se caractérise par une sur-estimation de l'éclairement sur les surfaces courbes. Ceci s'explique par le fait que la fenêtre est trop petite par rapport à la surface réelle sur laquelle les photons sont cherchés.
- Le *biais de frontière* est l'effet inverse : il s'agit d'une sous-estimation de l'éclairement sur le bord des objets. Sa cause est l'utilisation d'une fenêtre plus grande que la surface réelle sur laquelle les photons sont utilisés.
- Le *biais de proximité* n'est pas dû à un problème de domaine mais de photons utilisés dans l'estimation : certains photons sont utilisés à tort dans l'estimation. Il en résulte des débordements lumineux, particulièrement visibles sous les murs ou les cloisons fines dans les scènes architecturales.

Une première approche proposée par Jensen [Jensen 2001] consiste à ne pas utiliser directement la carte de photons au point d'impact x mais seulement après une réflexion indirecte. Cette approche est appelée *collecte finale*. La figure C.4 illustre le procédé. Cette méthode permet de repousser un cran plus loin le biais. En effet, plutôt qu'il n'apparaisse directement au premier impact, il est repoussé à la réflexion suivante. Le résultat est visuellement convainquant, bien que cette approche ne résolve pas le problème du biais (le biais de proximité est toujours visible sur le bord des cloisons par exemple). En outre, elle est très coûteuse du fait du nombre important de rayons de collecte à utiliser (un moteur de rendu commercial comme Mental Ray en utilise 1000 par point). Havran [Havran 2005b] et Christensen [Christensen 1999] accélèrent les requêtes de collecte finale, le premier en les ré-ordonnant et le second en les pré-calculant. Dans ce deuxième cas, les estimations de densité sont pré-calculées par photon ; dès lors la requête dans la carte de photons ne consiste plus qu'en la recherche du plus proche voisin. Bien qu'accelérant les requêtes, ces approches ne résolvent pas le problème du biais.

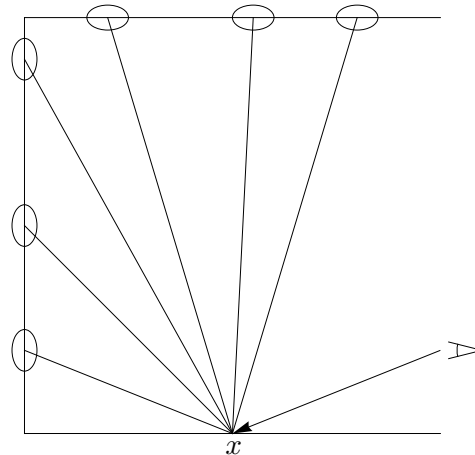


FIGURE C.4 – Exemple de collecte finale. Pour un point x donné, des rayons sont ré-émis dans la scène et à chaque point d'impact de ces rayons avec la scène, une estimation de densité est effectuée.

Lavignotte *et al.* [Lavignotte 2003], dans un contexte d'écrasement de photons (une méthode de rendu reposant également sur une estimation de densité), déterminent les surfaces utiles de l'estimation de densité (*i.e.* lorsqu'une estimation est effectuée au bord d'un objet, le bord limite la surface A^k). Lors de l'estimation de densité, l'aire de la surface sur laquelle sont utilisés les photons est calculée et le biais présent au bord des objets est alors supprimé. Cependant, cette correction n'est que partielle puisqu'elle repose seulement sur la géométrie des objets. La figure C.5 donne un exemple dans lequel la géométrie des objets n'est pas suffisante pour corriger le biais de frontière.

Schregle [Schregle 2003, Schregle 2005] propose une approche pour réduire le biais de proximité. Il réalise une série d'estimation de densité en faisant varier le nombre k de photons dans l'estimation entre deux bornes $[N_{min}, N_{max}]$. Pour chacune des estimations, une

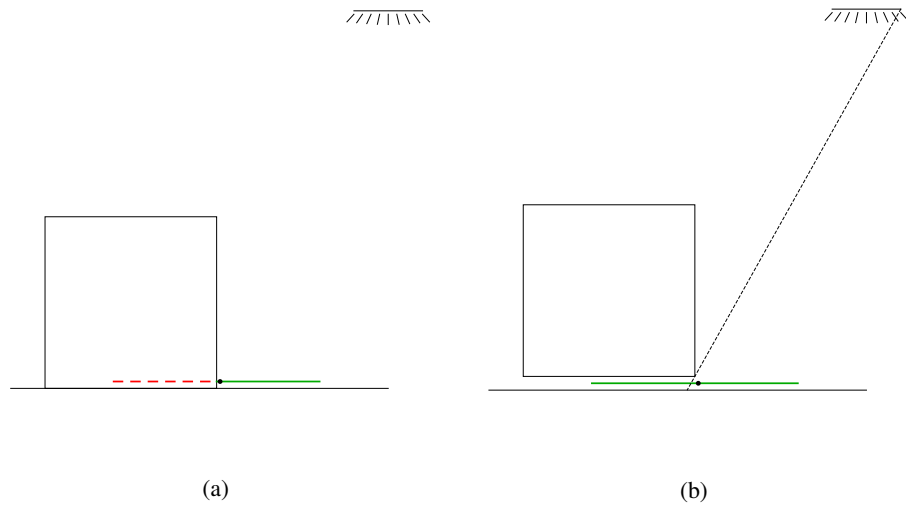


FIGURE C.5 – Exemple de calcul d'éclairage direct avec correction du biais de frontière par méthode de Lavignotte *et al.* (a) Le cube est posé sur le sol, le domaine (en vert) est bien tronqué. (b) Le cube est légèrement surélevé, la méthode proposée donne une aire trop importante (en pointillés, la limite de la zone de contribution de la source).

métrique évalue le rapport bruit *vs* biais dans l'estimation. Un critère statistique permet alors de sélectionner le meilleur choix pour le nombre k d'échantillons à utiliser dans l'estimation de densité. Cette méthode permet de diminuer le biais dans le sens où un rayon de recherche plus petit sera utilisé au bord des objets. Néanmoins le biais est toujours présent, le coût des estimations de densité successives pour le choix du paramètre k optimal demeure important, et le choix des bornes N_{min} et N_{max} reste empirique.

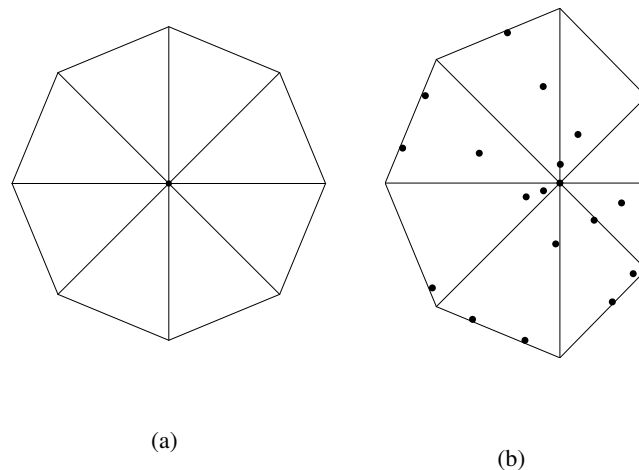


FIGURE C.6 – Exemple d'*octoboxes*. (a) Il s'agit d'une boîte englobante en dimension 2 possédant 8 cotés. (b) Les dimensions de la boîte sont adaptées aux photons collectés.

Tobler *et al.* [Tobler 2006] proposent deux solutions pour diminuer à la fois le biais de frontière et le biais de proximité dans les scènes architecturales. Ils utilisent une structure

appelée *octobox* permettant à la fois de filtrer les photons utilisés dans l'estimation de densité et de déterminer un domaine plus précis que celui de base. Une *octobox* est une boîte englobante régulière à huit côtés (*cf.* figure C.6(a)); les distances entre les sommets de la boîte et point d'estimation sont choisies de façon à s'adapter aux photons collectés (*cf.* figure C.6(b)). De plus, pour diminuer le biais de proximité, les auteurs proposent d'utiliser des rayons dans quatre directions principales de façon à limiter la taille des *octoboxes* et donc filtrer des photons (*cf.* figure C.7(a)). Malheureusement, comme le montrent les auteurs, cette méthode souffre d'instabilité lorsque peu de photons sont utilisés (les *octoboxes* deviennent très petites et souvent mal formées) et la limitation de la taille des boîtes par les quatre directions échoue dans certains cas (*cf.* figure C.7(b)).

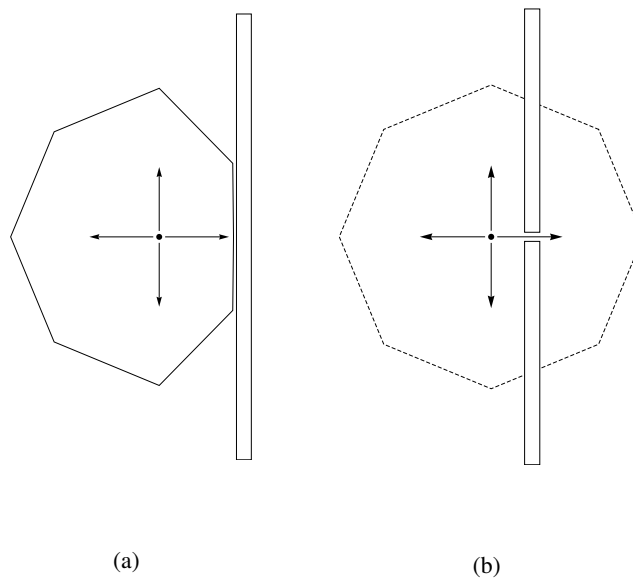


FIGURE C.7 – Exemple de réduction de la taille d'une *octobox* à partir des quatre directions principales. (a) La boîte est réduite à cause du mur de droite. (b) La réduction échoue à cause de l'ouverture dans le mur.

Pré-calculs de transfert énergétique

Cette annexe est extraite du chapitre 2 de la thèse de Romuald Perrot [Perrot 2012].

Lorsque l'objectif est la production interactive d'images de synthèse réalistes, une solution est de pré-calculer un maximum d'informations, afin de les exploiter rapidement lors du rendu final. Le pré-calcul représente une quantité importante de données et un stockage efficace est nécessaire (*i.e.* grande compression et rapidité d'exploitation). La principale contribution des méthodes de Pré-calcul de Transfert Énergétique (PTE) est de projeter l'ensemble des pré-calculs sur des bases de fonctions *ad hoc*.

Même si l'approche originale repose sur une base d'harmoniques sphériques, nous englobons dans le terme PTE toutes les méthodes comportant une projection des pré-calculs sur des bases de fonctions.

Cette annexe présente des bases de fonctions et quelques propriétés sur celles-ci. Ensuite, elle introduit la méthode PTE de Sloan *et al.*, et présente quelques extensions.

D.1 Bases de fonctions

Les méthodes PTE s'appuient sur les bases de fonctions pour stocker les pré-calculs de la simulation d'éclairage. En effet, toute fonction continue peut être représentée à l'aide d'une combinaison linéaire de fonctions de base. De plus, un signal peut être efficacement compressé et reconstruit à l'aide de deux opérations nommées *projection* et *reconstruction*.

Cette section présente quelques propriétés sur les bases orthonormales de fonctions.

D.1.1 Bases de fonctions orthonormales

Soit $\{b_1(x), b_2(x), b_3(x), \dots, b_n(x)\}$ une base de fonctions sur un domaine \mathcal{D} , pour $n \in \mathbb{N}^*$. Cette base est *orthogonale* si et seulement si :

$$\forall i, j \in [1 \dots n]^2, \quad \int_{\mathcal{D}} b_i(x)b_j(x) dx = \begin{cases} 0 & \text{si } i \neq j, \\ \neq 0 & \text{sinon.} \end{cases} \quad (\text{D.1})$$

De plus, elle est *orthonormale* si et seulement si :

$$\int_{\mathcal{D}} b_i(x)b_i(x) dx = 1, \forall i \in [1 \dots n]. \quad (\text{D.2})$$

D.1.2 Utilisation d'une base de fonctions orthonormales

Une fonction peut être représentée de manière exacte et unique par une combinaison linéaire des fonctions de base orthonormale. Le calcul des coefficients de cette combinaison est appelée *projection*. L'opération inverse, *i.e.* le calcul effectif de la fonction à partir des coefficients est appelée *reconstruction*.

La *projection* d'une fonction $f(x)$ sur une fonction $b_i(x)$ de la base consiste à calculer la valeur suivante :

$$c_1 = \int_{\mathcal{D}} f(x)b_i(x)dx. \quad (\text{D.3})$$

La reconstruction de la fonction à partir des n projections sur les fonctions de la base est donnée par :

$$f(x) = \sum_{i=1}^n c_i b_i(x). \quad (\text{D.4})$$

L'intégrale du produit de deux fonctions $f(x)$ et $g(x)$ peut être calculée directement à l'aide des coefficients de projection orthonormale de chacune des fonctions. Ainsi, si \tilde{f}_i et \tilde{g}_i (avec $i \in [1 \dots n]$) sont les projections respectives de $f(x)$ et $g(x)$ sur les fonctions de base alors l'intégrale du produit de $f(x)$ par $g(x)$ est :

$$\int_{\mathcal{D}} f(x)g(x) dx = \sum_{i=1}^n \tilde{f}_i \tilde{g}_i \quad (\text{D.5})$$

Cette propriété est vraie pour toute base de fonctions orthonormales. Si les fonctions de base sont seulement orthogonales, un coefficient supplémentaire doit être inclus dans le produit. En effet, cette identité se base sur le fait que le produit entre deux fonctions orthonormales de même degré vaut 1.

D.1.3 Application avec les Harmoniques Sphériques

Nous illustrons la projection, la reconstruction et l'intégration en utilisant la base de fonctions orthonormales reposant sur les harmoniques sphériques.

D.1.3.1 Harmoniques Sphériques

Les harmoniques sphériques reposent sur les *polynômes associés de Legendre*, définis de la manière suivante :

$$P_n = \frac{1}{2^n n!} \frac{d^n}{dx^n} ((x^2 - 1)^n).$$

Ces polynômes sont orthogonaux sur l'intervalle $[-1 \dots 1]$:

$$\int_{-1}^1 P_r(x)P_s(x) dx = \begin{cases} \frac{2}{2n+1} & \text{si } r = s, \\ 0 & \text{sinon.} \end{cases}$$

Cette définition permet directement de déduire que la suite des polynômes de Legendre forme une base orthogonale. Les *polynômes associés de Legendre* peuvent être définis à partir des polynômes de Legendre :

$$P_l^m(x) = (-1)^m \sqrt{(1-x^2)^m} \frac{d^m}{dx^m} (P_l(x)),$$

où l et m sont respectivement appelés *degré* et *ordre* du polynôme associé, P_l étant le $l^{\text{ème}}$ polynôme de Legendre. Les termes l et m sont des entiers respectant : $0 \leq m \leq l$. Les polynômes peuvent s'écrire à l'aide de la formule de Rodrigues :

$$P_l^m = \frac{(-1)^m}{2^l l!} \sqrt{(1-x^2)^m} \frac{d^{l+m}}{dx^{l+m}} (x^2-1)^l.$$

Notons qu'en posant $m = 0$, nous retrouvons les polynômes de Legendre. À l'image de ces derniers, les polynômes associés sont orthogonaux, pour un ordre m fixé, sur $[-1 \dots 1]$:

$$\int_{-1}^1 P_r^m(x) P_s^m(x) dx = \begin{cases} \frac{2(s+m)!}{(2s+1)(s-m)!} & \text{si } r = s, \\ 0 & \text{sinon.} \end{cases}$$

Le tableau D.1 donne les premiers polynômes associés de Legendre.

$l \backslash m$	0	1	2	3
0	1			
1	x	$-\sqrt{1-x^2}$		
2	$\frac{1}{2}(3x^2-1)$	$-3x\sqrt{1-x^2}$	$3(1-x^2)$	
3	$\frac{1}{2}(5x^3-3x)$	$\frac{3}{2}(1-5x^2)\sqrt{1-x^2}$	$15x(1-x^2)$	$-15\sqrt{(1-x^2)^3}$

TABLE D.1 – Liste des premiers polynômes associés $P_l^m(x)$ de Legendre.

Les harmoniques sphériques $Y_l^m(\theta, \phi)$ réelles, de paramètres entiers l et m avec $0 \leq l$ et $-l \leq m \leq l$, sont définies à partir des polynômes associés de Legendre :

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2} N_l^m \cos(m\phi) P_l^m \cos(\theta) & \text{si } m < 0, \\ N_l^0 P_l^0 \cos(\theta) & \text{si } m = 0, \\ \sqrt{2} N_l^{|m|} \sin(|m|\phi) P_l^{|m|} \cos(\theta) & \text{si } m > 0, \end{cases}$$

où P_l^j est un polynôme associé de Legendre, et N_l^j est le facteur de normalisation suivant :

$$N_l^j = \sqrt{\frac{(2l+1)(l-j)!}{4\pi(l+j)!}}.$$

Les harmoniques sphériques forment une base de fonctions orthonormales, de taille infinie. La figure D.1 illustre les harmoniques sphériques jusqu'au degré 3.

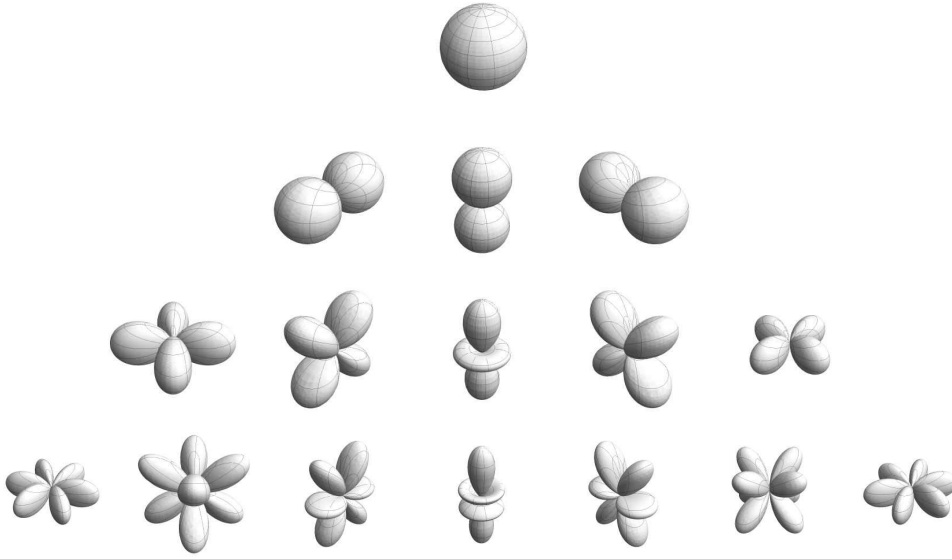


FIGURE D.1 – Exemples d’harmoniques sphériques pour $0 \leq l \leq 3$. Chaque ligne représente un degré l croissant et chaque colonne indique un ordre m croissant avec $-l \leq m \leq l$.

D.1.3.2 Projection, reconstruction et intégration à l’aide d’harmoniques sphériques

Comme toute base de fonctions, les harmoniques sphériques permettent de transformer un signal f en un ensemble de coefficients, afin d’en déterminer une approximation. Cette projection fonctionne sur un domaine sphérique :

$$c_l^m = \int_{\mathcal{S}^2} f(\theta, \phi) Y_l^m d(\theta, \phi),$$

où c_l^m est le coefficient associé à la fonction de base de degré l et d’ordre m , et avec comme domaine la sphère unité \mathcal{S}^2 .

La reconstruction permet de retrouver le signal f à partir des coefficients c_l^m . Il s’agit théoriquement de la somme infinie d’harmoniques sphériques :

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} c_l^m Y_l^m(\theta, \phi).$$

Dans la pratique, seuls les n premiers degrés sont utilisés :

$$\tilde{f}(\theta, \phi) = \sum_{l=0}^n \sum_{m=-l}^{m=l} c_l^m Y_l^m(\theta, \phi). \quad (\text{D.6})$$

La figure D.2 donne un exemple de reconstruction à l’aide d’harmoniques sphériques et en faisant varier le nombre n de fonctions de base utilisées, sur la fonction f_1 suivante :

$$f_1(\theta, \phi) = 1 + \frac{\sin(5\phi)}{5},$$

Lorsque peu de coefficients sont utilisés, *i.e.* lorsque seules les fonctions de base avec les degrés les plus faibles sont conservées, la fonction reconstruite est très éloignée du résultat attendu : seules les basses fréquences du signal sont reproduites. Au fur et à mesure de l'augmentation du degré, les signaux hautes fréquences sont de mieux en mieux reconstruits.

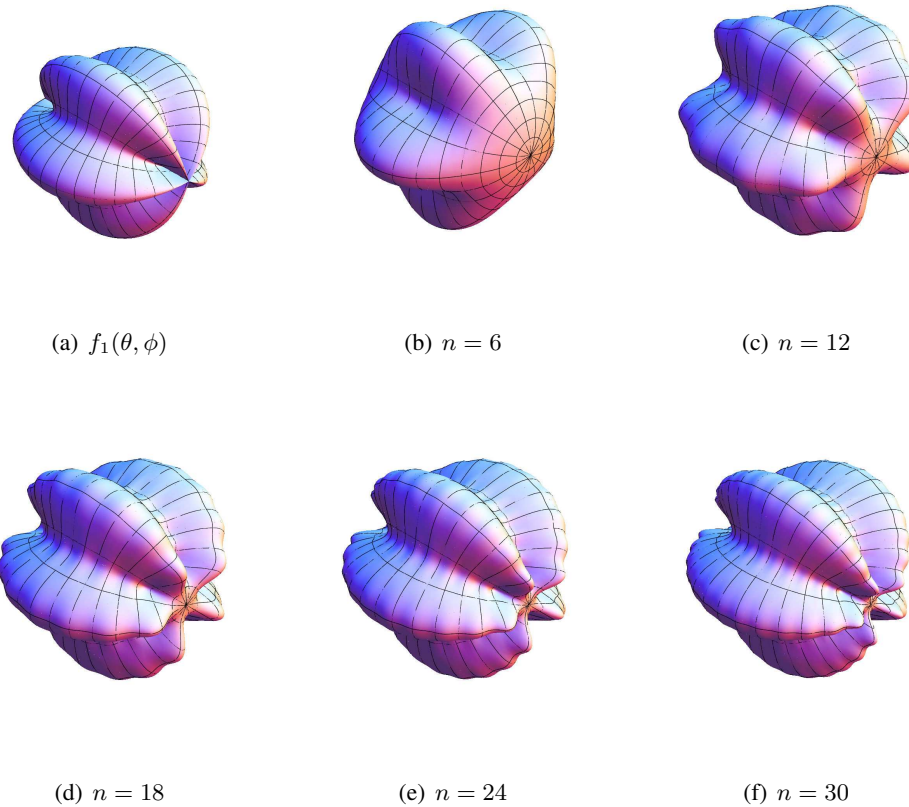


FIGURE D.2 – Exemple de reconstruction d'une fonction après projection sur une base de n harmoniques sphériques. (a) L'original. (b)-(f) Reconstruction en utilisant les n premiers degrés de la base d'harmoniques sphériques.

Soient maintenant deux fonctions f et g , et \tilde{f} et \tilde{g} leurs approximations respectives :

$$\tilde{f}(x) \approx \sum_{l=0}^n \sum_{m=-l}^l f_l^m Y_l^m(x),$$

$$\tilde{g}(x) \approx \sum_{i=0}^n \sum_{m=-l}^l g_l^m Y_l^m(x).$$

Les harmoniques sphériques formant une base de fonction orthonormales, l'intégrale du

produit des fonctions \tilde{f} et \tilde{g} s'exprime par la somme des produits de leurs coefficients :

$$\int_{\mathcal{S}^2} \tilde{f}(x) \tilde{g}(x) dx = \sum_{l=0}^n \sum_{m=-l}^l f_l^m g_l^m.$$

D.2 Méthodes PTE

Les méthodes PTE font quelques suppositions sur l'environnement :

- il est statique : les objets et les sources lumineuses ne se déplacent pas ;
- il est basse fréquence (des objets principalement mats) ;
- les sources lumineuses sont larges et proviennent généralement de cartes d'environnement.

Nous donnons une présentation des méthodes PTE à partir de l'ETL :

$$L_o(x, \mathbf{o}) = L_e(x, \mathbf{o}) + \int_{\mathcal{S}_x} L_i(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i.$$

Le terme émissif n'étant pas au coeur de l'étude, il est omis dans la suite. L'éclairement provient d'un environnement lointain, idéalement depuis une carte d'environnement ; la luminance incidente est alors réduite à :

$$L_i(x, \mathbf{i}) = L_i^e(\mathbf{i}) V(x, \mathbf{i}).$$

Le terme L_i^e représente la luminance incidente depuis la carte d'environnement, supposée à l'infini ; sa valeur ne dépend donc que de la direction \mathbf{i} . Le terme de visibilité permet de tester si la carte d'environnement est visible depuis le point x dans la direction \mathbf{i} . Dans un premier temps, et par souci de simplicité, seul l'éclairement direct est pris en compte (la luminance est alors notée L_0 pour indiquer qu'aucune réflexion indirecte n'est utilisée) :

$$L_0(x, \mathbf{o}) = \int_{\mathcal{S}_x} L_i^e(\mathbf{i}) V(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o}) (\mathbf{n}_x \cdot \mathbf{i}) d\omega_i.$$

Nous pouvons exprimer la contribution directe par une projection sur une base d'harmoniques sphériques :

$$L(\mathbf{i}) \approx \sum_{l=0}^n \sum_{m=-l}^l \alpha_l^m Y_l^m(\mathbf{i}). \quad (\text{D.7})$$

De plus, afin de simplifier les notations nous posons $j = (l, m)$:

$$L(\mathbf{i}) \approx \sum_{j=1}^{(n+1)^2} \alpha_j Y_j(\mathbf{i}),$$

où n est le degré maximal de l'harmonique sphérique. Avec la projection sur une base d'harmoniques sphériques l'ETL devient :

$$\begin{aligned} L_0(x, \mathbf{o}) &= \int_{\mathcal{S}_x} \sum_{j=1}^{(n+1)^2} \alpha_j Y_j(\mathbf{i}) V(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{i}) d\omega_{\mathbf{i}}, \\ &= \sum_{j=1}^{(n+1)^2} \alpha_j \int_{\mathcal{S}_x} Y_j(\mathbf{i}) V(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{i}) d\omega_{\mathbf{i}}. \end{aligned}$$

L'intégrale ne dépend plus de l'éclairement, et peut être pré-calculée :

$$T_{0,j}(x) = \int_{\mathcal{S}_x} Y_j(\mathbf{i}) V(x, \mathbf{i}) f(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{i}) d\omega_{\mathbf{i}}.$$

L'ETL se résume alors à un produit scalaire :

$$L_0(x, \mathbf{o}) = \sum_{j=1}^{(n+1)^2} \alpha_j T_{0,j}(x),$$

où $T_{0,j}(x)$ est appelé *coefficient de transport* (l'indice 0 indique un nombre de rebonds nul pour la contribution indirecte). De manière intuitive ce coefficient représente la façon dont l'objet projette les ombres sur lui même. Le rendu d'un point x de la scène est réalisé en deux étapes : récupérer les valeurs des coefficients d'éclairement et de transport puis effectuer un produit scalaire entre ces deux quantités. Ceci peut être réalisé de manière interactive sur carte graphique.

Ce calcul ne prend en compte que la contribution lumineuse directe. Afin de simuler les inter-réflexions diffuses, le même processus est évalué en prenant en compte le premier rebond puis le second et ainsi de suite. Par exemple pour la première réflexion, avec $y = \tau(x, \mathbf{i})$ l'ETL est :

$$L_1(x, \mathbf{o}) = \int_{\mathcal{S}} L_0(y, -\mathbf{i}) f_r(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{i})(1 - V(\mathbf{i})) d\omega_{\mathbf{i}}.$$

Le terme de visibilité est égal à $1 - V(\mathbf{i})$ puisque le premier rebond ne doit être pris en compte que si un objet est présent dans la direction \mathbf{i} . En exprimant $L_0(y, -\mathbf{i})$ de manière analogue à $L_0(x, \mathbf{i})$ (*i.e.* à l'aide d'un produit scalaire) l'équation pour la première réflexion est :

$$\begin{aligned} L_1(x, \mathbf{o}) &= \int_{\mathcal{S}} \left\{ \sum_{j=0}^{(n+1)^2} \alpha_j T_{0,j}(y) \right\} f_r(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{i})(1 - V(\mathbf{i})) d\omega_{\mathbf{i}}, \\ &= \sum_{j=0}^{(n+1)^2} \alpha_j \int_{\mathcal{S}} T_{0,j}(y) f_r(x, \mathbf{i}, \mathbf{o})(\mathbf{n}_x \cdot \mathbf{i})(1 - V(\mathbf{i})) d\omega_{\mathbf{i}}, \\ &= \sum_{j=0}^{(n+1)^2} \alpha_j T_{1,j}(x). \end{aligned}$$

Ainsi, en ne prenant en compte qu'un unique rebond indirect, la luminance au point x est :

$$\begin{aligned} L(x, \mathbf{o}) &= L_0(x, \mathbf{i}) + L_1(x, \mathbf{i}), \\ &= \sum_{j=0}^{(n+1)^2} \alpha_i (T_{0,j}(x) + T_{1,j}(x)). \end{aligned}$$

Le processus peut être répété pour autant de réflexions indirectes que désiré.

D.2.1 Pré-calcul et stockage

Le pré-calcul des coefficients de transport est effectué via une méthode de type Monte-Carlo pour l'ensemble des sommets des objets (ou parfois pour un ensemble plus dense si le nombre de points de l'objet est trop faible). Sloan *et al.* utilisent entre 10 000 et 30 000 directions pour échantillonner les sphères centrées en chacun de ces points. Les temps nécessaires à tous ces pré-calculs sont de l'ordre de quelques minutes à plusieurs heures lorsque plusieurs dizaines de milliers d'échantillons sont utilisés [Sloan 2002].

Pour chacun des échantillons une projection sur chacune des fonctions de base est calculée puis stockée. Sloan *et al.* utilisent 25 coefficients pour chacun des échantillons (*i.e.* ils utilisent les fonctions de base jusqu'au degré 4).

D.2.2 Limitations

Les limites de la méthode sont nombreuses, notamment le fait que l'éclairage et les FDRB ne peuvent être employées que pour des matériaux faiblement brillants. Ceci est dû au faible nombre de coefficients stockés. Pour utiliser des environnements hautes fréquences (*i.e.* sources plus petites projetant des ombres dures et FDRB plus brillantes), Ng *et al.* [Ng 2003] utilisent des *ondelettes*. Ces dernières permettent, à nombre de coefficients équivalent, d'approcher un signal ayant des hautes fréquences avec une plus grande précision par rapport à une base d'harmoniques sphériques. Cependant, l'utilisation d'un tel signal, comportant de très grandes variations, nécessite d'utiliser plus d'échantillons dans le pré-calcul. (Ng *et al.* en utilisent de l'ordre de 500 000). Ceci implique pour des scènes complexes des temps de pré-calculs de plusieurs heures, et un stockage pouvant dépasser l'ordre du Giga Octet. Des méthodes de compression basées sur des Analyses en Composantes Principales (ACP) [Sloan 2003] permettent de réduire l'espace nécessaire au stockage, au prix de temps de calculs encore plus longs.

La méthode PTE ne prend pas non plus en compte les environnements dynamiques. Afin de relâcher cette contrainte, Sloan *et al.* utilisent uniquement les *Harmoniques Zonales* (HZ), *i.e.* uniquement avec les ordres m nuls. Elles permettent la projection d'un signal symétrique par rapport à un axe du signal. Les auteurs indiquent comment obtenir le même processus pour un signal symétrique par rapport à un axe s quelconque. Ils décrivent également comment la rotation d'une approximation du signal peut être effectuée par le biais d'une rotation dans \mathbb{R}^3 , autorisant un calcul en $\mathcal{O}(n)$ au lieu de $\mathcal{O}(n^3)$. La possibilité d'un calcul rapide permet une rotation des coefficients de transfert et donc l'utilisation de modèles

déformables. Néanmoins, cette approche ne permet pas de gérer correctement les ombres projetées par un objet sur lui-même puisque ceci nécessite de re-calculer complètement les coefficients de transfert.

Bibliographie

- [Adrià 2009] F. Adrià, S. Pattanaik, B. Carles et X. Pueyo. *BRDFLab : A general system for designing BRDFs*. In Proceedings CEIG'09. Eurographics, 2009. (Cité en page 85.)
- [Agelet 2000] F. A. Agelet, A. Formella, J. M. H. Rabanos, F. I. de Vicente et F. P. Fontan. *Efficient Ray-Tracing Acceleration Techniques for Radio Propagation Modeling*. IEEE-VEH, vol. 49, pages 2089–2104, November 2000. (Cité en page 101.)
- [Aila 2009] T. Aila et S. Laine. *Understanding the Efficiency of Ray Traversal on GPUs*. In Proc. High-Performance Graphics 2009, pages 145–149, 2009. (Cité en page 88.)
- [Aila 2012] T. Aila, S. Laine et T. Karras. *Understanding the Efficiency of Ray Traversal on GPUs – Kepler and Fermi Addendum*. NVIDIA Technical Report NVR-2012-02, NVIDIA Corporation, Juin 2012. (Cité en pages 88, 91 et 92.)
- [Airey 1990] J. M. Airey, J. H. Rohlf et F. P. J. Brooks. *Towards image realism with interactive update rates in complex virtual building environments*. SIGGRAPH Comput. Graph., vol. 24, no. 2, pages 41–50, Février 1990. (Cité en page 88.)
- [Akenine-Möller 2002] T. Akenine-Möller et U. Assarsson. *Approximate soft shadows on arbitrary surfaces using penumbra wedges*. In EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering, pages 297–306, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. (Cité en page 63.)
- [Amanatides 1984] John Amanatides. *Ray tracing with cones*. In SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pages 129–135, New York, NY, USA, 1984. ACM Press. (Cité en pages 63 et 88.)
- [Andres 1997] E. Andres, R. Acharya et C. Sibata. *Discrete Analytical Hyperplanes*. Graphical Models And Image Processing, vol. 59, no. 5, pages 302–309, Septembre 1997. (Cité en pages 101 et 128.)
- [Apostu 2012] O. L. Apostu, F. Mora, D. Ghazanfarpour et L. [Aveneau](#). *Analytic Ambient Occlusion using Exact from-Polygon Visibility*. Computer & Graphics, vol. 36, no. 6, pages 727–739, Octobre 2012. (Cité en pages 58, 78 et 82.)
- [Appel 1968] Arthur Appel. *Some techniques for shading machine renderings of solids*. In AFIPS Spring Conference, pages 37–45, 1968. (Cité en page 88.)
- [Aronov 1993] B. Aronov, M. Pellegrini et M. Charir. *On the zone of a surface in a hyper-plane arrangement*. Discrete Computational Geometry, vol. 9, no. 2, pages 177–186, 1993. (Cité en page 29.)
- [Athanasiadou 2000] G. E. Athanasiadou et A. R. Nix. *A Novel 3-D Indoor Ray-Tracing Propagation Model : The Path Generator and Evaluation of Narrow-Band and Wide-Band Predictions*. IEEE-VEH, vol. 49, pages 1152–1168, July 2000. (Cité en page 101.)

- [Atherton 1978] P. Atherton, K. Weiler et D. Greenberg. *Polygon shadow generation*. In SIGGRAPH '78 : Proceedings of the 5th annual conference on Computer graphics and interactive techniques, pages 275–281, New York, NY, USA, 1978. ACM Press. (Cité en page 63.)
- [Audin 2004] Michèle Audin. *Géométrie*. EDP Sciences, 2004. (Cité en page 45.)
- [Aveneau 2000] L. Aveneau, Y. Pousset, R. Vauzelle et M. Mériaux. *Development and Evaluations of Physical and Computer Optimizations for the 3D UTD Model*. Davos, Switzerland, April 2000. AP'2000. (Cité en pages 104 et 106.)
- [Aveneau 2002] Lilian Aveneau. *TOPASE : manuel utilisateur*, Février 2002. Rapport de Contrat de Recherche Externe n°01-1B-323, France Télécom, 21 pages. (Cité en pages 38 et 100.)
- [Aveneau 2003a] L. Aveneau, P. Combeau, R. Vauzelle et M. Mériaux. *Efficient computation of radio coverage zone using a spatial partitionment approach*. In Proceedings of IEEE VTC, volume 4, pages 2421–2425, Orlando, États-Unis, Octobre 2003. (Cité en pages 39, 104 et 107.)
- [Aveneau 2003b] L. Aveneau et F. Mora. *Optimisation du temps de calcul du logiciel à tracé de rayons TRIO, Rapport à TO+2*, Septembre 2003. Rapport de Contrat de Recherche Externe n°42.538.233, France Télécom, 20 pages. (Cité en page 108.)
- [Aveneau 2004a] L. Aveneau et P. Combeau. *Path Finding Based on Monte Carlo Techniques Compared with a Full Ray-Tracing Approach in Narrow and Wide Bands*. Milan, Mai 2004. VTC2004. (Cité en pages 104 et 107.)
- [Aveneau 2004b] L. Aveneau et F. Mora. *Bibliothèque SIC_Graphe.DLL, Manuel utilisateur, Rapport à TO+15*, Octobre 2004. Rapport de Contrat de Recherche Externe n°42.538.233, France Télécom, 367 pages. (Cité en page 108.)
- [Aveneau 2004c] L. Aveneau et F. Mora. *Optimisation du temps de calcul du logiciel à tracé de rayons TRIO, Rapport à TO+6*, Janvier 2004. Rapport de Contrat de Recherche Externe n°42.538.233, France Télécom, 33 pages. (Cité en page 108.)
- [Aveneau 2004d] L. Aveneau et F. Mora. *Optimisation du temps de calcul du logiciel à tracé de rayons TRIO, Rapport à TO+9*, Avril 2004. Rapport de Contrat de Recherche Externe n°42.538.233, France Télécom, 34 pages. (Cité en page 108.)
- [Aveneau 2005] L. Aveneau et F. Mora. *Optimisation du temps de calcul du logiciel à tracé de rayons TRIO, Rapport Final*, Janvier 2005. Rapport de Contrat de Recherche Externe n°42.538.233, France Télécom, 34 pages. (Cité en page 108.)
- [Aveneau 2009] Lilian Aveneau. *Les coordonnées de Plücker revisitées*. Revue Electronique Francophone de l'Informatique Graphique, vol. 3, no. 2, pages pages 59–68, Janvier 2009. (Cité en pages 16 et 22.)
- [Aveneau 2011] L. Aveneau, S. Charneau, L. Fuchs et F. Mora. *A Framework for n-dimensional Visibility Computations*. In Leo Dorst et Joan Lasenby, éditeurs, Guide to Geometric Algebra in Practice, pages 273–294. Springer London, Septembre 2011. (Cité en pages 50, 52, 54 et 55.)

- [Aveneau 2012] L. Aveneau, E. Andres et L. Fuchs. *Lines Classification in the Conformal Space $\mathbb{R}^{n+1,1}$* . In AGACSE'2012, La Rochelle, France, Juillet 2012. (Cité en page 128.)
- [Bajaj 1996] C. L. Bajaj et V. Pascucci. *Splitting a complex of convex polytopes in any dimension*. In ACM, editeur, Proceedings of the 12th Annual Symposium on Computational Geometry, pages 88–97, May 1996. (Cité en page 32.)
- [Balanis 1989] C. Balanis. *Advanced Engineering Electromagnetics*. Edition Willey, 1989. (Cité en pages 7 et 97.)
- [Barry 1993] J. R. Barry, J. M. Kahn, W. J. Krause, E. A. Lee et D. G. Messerschmitt. *Simulation of multipath Impulse Response for Indoor Wireless Optical Channels*. IEEE Journal on Selected Areas in Communications, vol. 11, no. 3, Avril 1993. (Cité en pages 120 et 121.)
- [Bass 1972] J. Bass et C. Do. Algèbre, volume 1 of *Mathématiques*. Masson & C^{ie}, 1972. (Cité en page 38.)
- [Bekaert 2003] P. Bekaert, P. Slusallek, R. Cools, V. Havran et H.-P. Seidel. *A custom designed density estimation method for light transport*. Rapport technique MPI-I-2003-4-004, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, Septembre 2003. (Cité en page 74.)
- [Bentley 1975] Jon Louis Bentley. *Multidimensional binary search trees used for associative searching*. Commun. ACM, vol. 18, no. 9, pages 509–517, Septembre 1975. (Cité en pages 88 et 100.)
- [Bergeron 1986] W. Bergeron. *A General Version of Crow'Shadow Volumes*. IEEE Computer Graphics and applications, vol. 6, no. 9, pages 17–28, 1986. (Cité en page 63.)
- [Bittner 2002] Jiří Bittner. *Hierarchical Techniques for Visibility Computations*. PhD thesis, Czech Technical University in Prague, October 2002. (Cité en pages 12, 27, 29, 31, 33, 53, 55 et 66.)
- [Boissonnat 1998] J-D. Boissonnat et M. Yvinec. *Algorithmic geometry*. Cambridge University Press, UK, 1998. Translated by Hervé Brönnimann. (Cité en page 26.)
- [Bouatouch 1992] K. Bouatouch et P. Tellier. *A two-pass physics-based global lighting model*. In Graphics Interface 92, May 11-15, pages 319–328, 1992. (Cité en page 100.)
- [Bredow 2002] Rob Bredow. *Renderman on Film*. SIGGRAPH 2002 Course Notes Course, vol. 16, no. 6, 2002. (Cité en page 78.)
- [Brouillat 2008] J. Brouillat, P. Gautron et K. Bouatouch. *Photon-driven Irradiance Cache*. Computer Graphics Forum, vol. 27, no. 7, pages 1971–1978, 2008. (Cité en page 163.)
- [Bunnell 2005] M. Bunnell. *Dynamic Ambient Occlusion and Indirect Lighting*. In Matt Pharr et Randima Fernando, editeurs, GPU Gems 2, pages 223–233. Addison-Wesley Professional, 2005. (Cité en page 79.)
- [Cagnac 1965] G. Cagnac, E. Ramis et J. Commeau. *Nouveau cours de mathématiques spéciales : Géométrie*, volume 3. Masson & C^{ie}, 1965. (Cité en page 45.)

- [Casanova 1965] Gaston Casanova. Géométrie analytique, volume 3 of *Mathématiques Spéciales*. Librairie Belin, 1965. (Cité en pages 45 et 47.)
- [Charneau 2005] S. Charneau, L. Fuchs et L. [Avenueau](#). *Implantation de l'algèbre géométrique en Objective Caml*. In Actes des Journées Francophones pour les Langages Applicatifs 2005, Obernai (Contreforts des Vosges), mars 2005. (Cité en page 50.)
- [Charneau 2007a] S. Charneau, L. [Avenueau](#) et L. Fuchs. *Exact, robust and efficient full visibility computation in the Plücker space*. *Visual Computer*, vol. 23, no. 9-11, pages pages 773–782, Septembre 2007. (Cité en pages 55, 56, 57 et 70.)
- [Charneau 2007b] Sylvain Charneau. *Étude et application des algèbres géométriques pour le calcul de la visibilité globale dans un espace projectif de dimension $n \geq n$* . PhD thesis, Université de Poitiers, Décembre 2007. (Cité en pages 37, 55 et 56.)
- [Chen 1996] S. H. Chen et S. K. Jeng. *SBR image approach for radio wave propagation in tunnels with and without traffic*. *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pages 570–578, 1996. (Cité en page 116.)
- [Chen 2010] L.-H. Chen, T.-C. Tsai et Y.-S. Chen. *Grouped photon mapping*. *The Visual Computer*, vol. 26, pages 217–226, 2010. (Cité en page 74.)
- [Chin 1989] N. Chin et S. Feiner. *Near real-time shadow generation using BSP trees*. In SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques, pages 99–106, New York, NY, USA, 1989. ACM Press. (Cité en page 63.)
- [Chin 1992] N. Chin et S. Feiner. *Fast object-precision shadow generation for area light sources using BSP trees*. In SI3D '92 : Proceedings of the 1992 symposium on Interactive 3D graphics, pages 21–30, New York, NY, USA, 1992. ACM Press. (Cité en page 63.)
- [Christensen 1999] Per Henrik Christensen. *Faster photon map global illumination*. *Journal of Graphics Tools*, vol. 4, pages 1–10, 1999. (Cité en page 165.)
- [Clifford 1876] William Kingdon Clifford. *Mathematical Papers*. edited by Mac Millan, 1876. (Cité en page 40.)
- [Cocheril 2005a] Y. Cocheril, R. Vauzelle, L. [Avenueau](#) et M. Khoudeir. *Modélisation et Prise en Compte des Surfaces Rugueuses pour un Simulateur de Propagation d'Ondes à 60 GHz en Milieu Indoor*. In Actes des 14ème Journées Nationales Micro-ondes, Nantes (France), Mai 2005. (Cité en page 105.)
- [Cocheril 2005b] Y. Cocheril, R. Vauzelle, L. [Avenueau](#) et M. Khoudeir. *Rough Surfaces Influence on an Indoor Propagation Simulation at 60 GHz*. In ECPS '05 Proceedings, Brest (France), March 2005. IEEE. (Cité en page 105.)
- [Cocheril 2006] Y. Cocheril, R. Vauzelle et L. [Avenueau](#). *3D Channel Simulations Including Scattering from non-Gaussian Rough Surfaces*. In VTC '06 Fall Proceedings, Montreal (Canada), Septembre 2006. IEEE. (Cité en page 105.)
- [Combeau 2002a] P. Combeau, L. [Avenueau](#) et R. Vauzelle. *Optimisation en temps de calcul pour la prédiction de zones de couverture : méthode TOPASE*, Mars 2002. Rapport

- final de Contrat de Recherche Externe n°01-1B-323, France Télécom, 65 pages. (Cité en page 100.)
- [Combeau 2002b] P. Combeau, R. Vauzelle, L. [Aveneau](#) et Y. Pousset. *An acceleration technique for the radioelectric coverage prediction in small and microcell configurations*. In European Workshop on Integrated Radiocommunicated Systems, Angers, France, Mai 2002. (Cité en page 39.)
- [Combeau 2002c] P. Combeau, R. Vauzelle, L. [Aveneau](#) et Y. Pousset. *Une technique d'accélération pour la prédiction de zone de couverture radioélectrique*. In Proceedings of PAA, Rennes, France, Mars 2002. (Cité en page 39.)
- [Combeau 2003] P. Combeau, R. Vauzelle, L. [Aveneau](#) et Y. Pousset. *Une optimisation en temps de calcul pour la prédiction de zones de couverture radioélectriques*. Annales des Télécommunications, vol. 58, no. 7-8, pages 1130–1156, 2003. (Cité en page 39.)
- [Combeau 2004a] P. Combeau, L. [Aveneau](#), R. Vauzelle et C. Chatellier. *Deterministic propagation model influence on a wireless digital transmission simulation in real environment*. In Proceedings of IEEE PIMRC 2004, volume 4, pages 2421–2425, Barcelonne, Espagne, Septembre 2004. (Cité en page 39.)
- [Combeau 2004b] Pierre Combeau. *Simulation efficace et caractérisation du canal radiomobile en environnement réel. Application aux systèmes sans fil*. PhD thesis, Université de Poitiers, novembre 2004. (Cité en pages 37 et 38.)
- [Combeau 2006] P. Combeau, L. [Aveneau](#), R. Vauzelle et Y. Pousset. *Efficient 2D ray-tracing method for narrow and wide-band channel characterization in micro-cellular configurations*. IEE Proceedings. Microwaves, Antennas and Propagation, vol. 153, no. 6, pages 502–509, Décembre 2006. (Cité en pages 39 et 104.)
- [Combeau 2007] P. Combeau, R. Vauzelle, Y. Pousset et L. [Aveneau](#). *An optimization in computation time for the prediction of radio coverage zones*. Radio Science, vol. 42, page RS1003, Février 2007. (Cité en pages 39 et 104.)
- [Combeau 2012] P. Combeau, L. [Aveneau](#) et A. Behloul. *Modélisation et simulation efficaces du canal de propagation optique sans fil en environnement réel*. In Proceedings du GDR Ondes - Journées thématiques Modélisation du visible au THz & Plasmonique moléculaire et hybride, France, Novembre 2012. (Cité en page 133.)
- [Cook 1984] R.L. Cook, T. Porter et L. Carpenter. *Distributed ray tracing*. In SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pages 137–145, New York, NY, USA, 1984. ACM Press. (Cité en pages 63 et 88.)
- [Cook 1986] R. L. Cook. *Stochastic sampling in computer graphics*. ACM Trans. Graph., vol. 5, no. 1, pages 51–72, 1986. (Cité en page 63.)
- [Crow 1977] Franklin C. Crow. *Shadow algorithms for computer graphics*. In SIGGRAPH '77 : Proceedings of the 4th annual conference on Computer graphics and interactive techniques, pages 242–248, New York, NY, USA, 1977. ACM Press. (Cité en page 63.)

- [Delachet 1964] André Delachet. *La géométrie projective. Que sais-je.* Presses Universitaires de France, 1964. (Cité en page 15.)
- [Didascalou 2000] D. Didascalou, T. M. Schafer, F. Weinmann et W. Wiesbeck. *Ray-density normalization for ray-optical wave propagation modeling in arbitrarily shaped tunnels.* IEEE Transactions on Antennas and Propagation, vol. 48, no. 9, pages 1316–1325, 2000. (Cité en page 116.)
- [Didascalou 2001] D. Didascalou, J. Maurer et W. Wiesbeck. *Subway tunnel guided electromagnetic wave propagation at mobile communications frequencies.* IEEE Transactions on Antennas and Propagation, vol. 49, no. 11, pages 590–1596, 2001. (Cité en pages 116 et 118.)
- [Dorst 2007] L. Dorst, D. Fontijne et S. Mann. *Geometric algebra for computer science : An object oriented approach to geometry.* Morgan Kaufmann Publishers, 2007. (Cité en pages 40 et 50.)
- [Duguet 2002] F. Duguet et G. Drettakis. *Robust epsilon visibility.* In SIGGRAPH'02 : Proceedings of the 29th annual Conference on Computer Graphics and Interactive Techniques, pages 567–575, New York, USA, 2002. ACM Press. (Cité en page 11.)
- [Durand 1997a] F. Durand, G. Drettakis et C. Puech. *The 3d visibility complexe : a unified data structure for global visibility of scenes of polygons and smooth objects.* In Canadian Conference on Computational Geometry, August 1997. (Cité en page 10.)
- [Durand 1997b] F. Durand, G. Drettakis et C. Puech. *The visibility skeleton : a powerful and efficient multi-purpose global visibility tool.* Computer Graphics, vol. 31, no. 3A, pages 89–100, August 1997. (Cité en page 11.)
- [Durand 1999a] F. Durand, G. Drettakis et C. Puech. *Fast and accurate hierarchical radiosity using global visibility.* ACM Transaction on Graphics, April 1999. (Cité en page 11.)
- [Durand 1999b] Frédo Durand. *Visibilité tridimensionnelle : étude analytique et applications.* PhD thesis, Université Joseph Fourier, iMAGIS-GRAVIR/IMAG-INRIA, juillet 1999. (Cité en page 18.)
- [Durand 2000] Frédo Durand. *A Multidisciplinary Survey of Visibility.* Siggraph'2000 courses notes on Visibility, 2000. (Cité en page 9.)
- [Dutre 1996] Philip Dutre. *Mathematical Frameworks and Monte Carlo Algorithms for Global Illumination in Computer Graphics.* PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, Septembre 1996. (Cité en page 155.)
- [Escarieu 2001] F. Escarieu, Y. Pousset, L. [Aveneau](#) et R. Vauzelle. *Outdoor and Indoor Channel Characterization by a 3D Simulation Software.* PIMRC, San Diego, USA, 2001. (Cité en page 104.)
- [Everitt 2003] C. Everitt et M. J. Kilgard. *Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering,* 2003. (Cité en page 63.)
- [Foco 2003] M. Foco, P. Polotti, A. Sarti et S. Tubaro. *Sound spatialization based on fast beam tracing in the dual space.* In Proceedings of the 6th International Conference on Digital Audio Effects, London, UK, September 2003. (Cité en page 13.)

- [Foley 2005] T. Foley et J. Sugerma. *KD-tree acceleration structures for a GPU raytracer*. In Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, HWWS '05, pages 15–22, 2005. (Cit  en page 88.)
- [Fontijne 2003] D. Fontijne, L. Dorst et T. Bouma. *Gaigen : A Geometric Algebra Implementation Generator*. <http://www.science.uva.nl/ga/gaigen/index.html>, 2003. University of Amsterdam. (Cit  en page 50.)
- [Fontijne 2007] D. Fontijne et L. Dorst. *Gaigen2*. <http://staff.science.uva.nl/~fontijne/gaigen2.html>, 2007. University of Amsterdam. (Cit  en page 50.)
- [Forest 2006] V. Forest, L. Barthe et M. Paulin. *Realistic Soft Shadows by Penumbra-Wedges Blending*. In Graphics Hardware (GH), Vienne, Autriche, 03/09/2006-04/09/2006, pages 39–48, <http://www.eg.org/>, 2006. Eurographics. (Cit  en page 63.)
- [Forest 2008] V. Forest, L. Barthe et M. Paulin. *Accurate Shadows by Depth Complexity Sampling*. Computer Graphics Forum, Eurographics 2008 Proceedings, vol. 27, no. 2, pages 663–674, 2008. (Cit  en page 63.)
- [Forest 2009] V. Forest, L. Barthe, G. Guennebaud et M. Paulin. *Soft textured shadow volume*. Computer Graphics Forum, Eurographics Symposium on Rendering 2009, vol. 28, no. 4, pages 1111–1121, 2009. (Cit  en page 63.)
- [Fradin 2005] D. Fradin, D. Meneveaux et S. Horna. *Out-of-core Photon-Mapping for Large Buildings*. In Proceedings of Eurographics symposium on Rendering, June 2005. (Cit  en page 88.)
- [Fuchs 1980] H. Fuchs, Z. M. Kedem et B. F. Naylor. *On visible surface generation by a priori tree structures*. SIGGRAPH Comput. Graph., vol. 14, no. 3, pages 124–133, Juillet 1980. (Cit  en page 88.)
- [Fujimoto 1986] A. Fujimoto, T. Tanaka et K. Iwata. *ARTS : accelerated ray-tracing system*. IEEE Comput. Graph. Appl., vol. 6, no. 4, pages 16–26, Avril 1986. (Cit  en pages 88 et 100.)
- [Fyffe 2009] Graham Fyffe. *Cosine lobe based relighting from gradient illumination photographs*. In SIGGRAPH '09 : Posters, pages 80 :1–80 :1, New York, NY, USA, 2009. ACM. (Cit  en page 83.)
- [Ghazanfarpour 1998] D. Ghazanfarpour et J.-M. Hasenfratz. *A Beam Tracing with Precise Antialiasing for Polyhedral Scenes*. Computer Graphics, vol. 22, no. 1, pages 103–115, 1998. (Cit  en pages 6, 63 et 88.)
- [Glassner 1984] Andrew S. Glassner. *Space Subdivision for Fast Ray Tracing*. IEEE Computer Graphics and Applications, vol. 4, no. 10, pages 15–22, Octobre 1984. (Cit  en page 100.)
- [Glassner 1989] Andrew S. Glassner, editeur. *An introduction to ray tracing*. Academic Press Ltd., London, UK, UK, 1989. (Cit  en page 88.)
- [GNU 2006] GNU. *The GNU Multiple Precision Arithmetic Library*. <http://gmpilib.org/>, 2006. (Cit  en page 55.)

- [Goaoc 2004] Xavier Goaoc. *Structures de visibilité globales : tailles, calculs et dégénérescences*. PhD thesis, Université Nancy 2, Mai 2004. (Cité en page 11.)
- [Grassmann 1844] Hermann Günther Grassmann. *Die lineale Ausdehnungslehre*, Leipzig : Wiegand 1844. (Cité en page 40.)
- [Grassmann 1862] Hermann Günther Grassmann. *Die ausdehnungslehre. vollständig und in strenger form begründet*. Berlin : Enslin., 1862. English translation, 2000, by Lloyd Kannenberg, *Extension Theory*. American Mathematical Society. (Cité en page 40.)
- [Gritzmann 2005] P. Gritzmann et T. Theobald. *On the complexity of visibility problems with moving viewpoints*. In *Combinatorial and Computational Geometry*, volume 52, pages 377–397. MSRI publications, Cambridge University Press, 2005. (Cité en page 11.)
- [Günther 2007] J. Günther, S. Popov, H.-P. Seidel et P. Slusallek. *Realtime Ray Tracing on GPU with BVH-based Packet Traversal*. In *Proceedings of the IEEE/Eurographics Symposium on Interactive Ray Tracing 2007*, pages 113–118, sep 2007. (Cité en page 88.)
- [Haines 2000] Eric Haines. *A shaft culling tool*. *J. Graph. Tools*, vol. 5, no. 1, pages 23–26, Janvier 2000. (Cité en page 70.)
- [Hamilton 1853] William Rowan Hamilton. *Lectures on quaternions*. Hodges and Smith, Grafton Street, 1853. (Cité en page 40.)
- [Hammersley 1967] J. M. Hammersley et D. C. Handscomb. *Monte carlo methods*. Chapman and Hall, 1967. (Cité en page 145.)
- [Haumont 2005] D. Haumont, O. Mäkinen et S. Nirenstein. *A Low Dimensional Framework for Exact Polygon-to-Polygon Occlusion Queries*. In *Proceedings of the Eurographics Symposium on Rendering*, pages 211–222, June 2005. (Cité en pages 32, 33, 34, 53 et 55.)
- [Havran 1999] V. Havran et F. Sixta. *Comparison of Hierarchical Grids*. *Ray Tracing News*, vol. 12, no. 1, Juin 1999. (Cité en page 88.)
- [Havran 2001] V. Havran. *Heuristic Ray Shooting Algorithms*. PhD thesis, Czech Technical University in Prague, 2001. (Cité en page 66.)
- [Havran 2005a] V. Havran, J. Bittner, R. Herzog et H.-P. Seidel. *Ray Maps for Global Illumination*. In *Eurographics Workshop on Rendering Techniques*, pages 43–54, 2005. (Cité en page 74.)
- [Havran 2005b] V. Havran, R. Herzog et H.-P. Seidel. *Fast Final Gathering via Reverse Photon Mapping*. In *The European Association for Computer Graphics 26th Annual Conference : EUROGRAPHICS 2005*, pages 323–333, Dublin, Ireland, Août 2005. Blackwell. (Cité en page 165.)
- [Heckbert 1984] P. S. Heckbert et P. Hanrahan. *Beam tracing polygonal objects*. In *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 119–127, New York, NY, USA, 1984. ACM Press. (Cité en pages 63 et 88.)

- [Hegeman 2006] K. Hegeman, S. Premoze, M. Ashikhmin et G. Drettakis. *Approximate Ambient Occlusion For Trees*. In C. Sequin et M. Olano, éditeurs, Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, March 2006. (Cité en page 79.)
- [Heidmann 1991] Tim Heidmann. *Real Shadows, Real Time*. Iris Universe, vol. 18, pages 28–31, 1991. Silicon Graphics, Inc. (Cité en page 63.)
- [Hestenes 1966] David Hestenes. Spacetime algebra. Gordon & Breach, 1966. (Cité en page 40.)
- [Hestenes 1984] D. Hestenes et G. Sobczyk. Clifford algebra to geometric calculus. D. Reidel Publ. Co., Dordrecht, Boston, 1984. (Cité en page 40.)
- [Hestenes 1986] David Hestenes. New foundations for classical mechanics. D. Reidel Publ. Co., Dordrecht, Boston, 1986. (Cité en page 40.)
- [Hestenes 2001] David Hestenes. *Old Wine in New Bottles : A new algebraic framework for computational geometry*. In E. Bayro-Corrochano & G. Sobczyk, éditeur, Advances in Geometric Algebra with Applications in Science and Engineering, pages 1–14. Birkhäuser, 2001. (Cité en page 40.)
- [Hoferock 2007] J. Hoferock et Y. Jia. *High-Quality Ambient Occlusion*. In Hubert Nguyen, éditeur, GPU Gems 3, pages 257–274. Addison-Wesley Professional, 2007. (Cité en page 79.)
- [Horn 2007] D. R. Horn, J. Sugerman, M. Houston et P. Hanrahan. *Interactive k-d tree GPU raytracing*. In Proceedings of the 2007 symposium on Interactive 3D graphics and games, I3D '07, pages 167–174. ACM, 2007. (Cité en page 88.)
- [Horna 2009] S. Horna, D. Meneveaux, G. Damiani et Y. Bertrand. *Consistency constraints and 3D building reconstruction*. Computer-Aided Design, vol. 41, no. 1, pages 13–27, Janvier 2009. (Cité en page 88.)
- [Ize 2008] T. Ize, I. Wald et S.G. Parker. *Ray tracing with the BSP tree*. In Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on, pages 159–166, 2008. (Cité en page 130.)
- [Izenman 2008] A. Izenman. Modern Multivariate Statistical Techniques, Regression, Classification, and Manifold Learning. Springer New-York, 2008. (Cité en page 72.)
- [Jensen 2001] H. W. Jensen. Realistic image synthesis using photon mapping. A. K. Peters, Ltd., Natick, MA, USA, 2001. (Cité en pages 74, 164 et 165.)
- [Jones 1993] M.C. Jones. *Simple boundary correction for kernel density estimation*. Statistics and Computing, vol. 3, no. 3, pages 135–146, 1993. (Cité en page 73.)
- [Kajiya 1986] James T Kajiya. *The rendering equation*. SIGGRAPH Comput. Graph., vol. 20, no. 4, pages 143–150, 1986. (Cité en pages 142 et 153.)
- [Kay 1986] T. L. Kay et J. T. Kajiya. *Ray tracing complex scenes*. SIGGRAPH Comput. Graph., vol. 20, no. 4, pages 269–278, Août 1986. (Cité en page 88.)
- [Keller 1962] J. B. Keller. *Geometrical Theory of Diffraction*. J. Opt. Soc. Amer, vol. 52, pages 116–130, 1962. (Cité en page 98.)

- [Kirk 2007] A. G. Kirk et O. Arikan. *Real-time ambient occlusion for dynamic character skins*. In Proceedings of the 2007 symposium on Interactive 3D graphics and games, I3D '07, pages 47–52, New York, USA, 2007. ACM. (Cité en page 79.)
- [Koenderink 1976] J. J. Koenderink et A. van Doorn. *The singularities of visual mapping*. Biological Cybernetics, vol. 24, no. 1, pages 51–59, 1976. (Cité en page 10.)
- [Koenderink 1979] J. J. Koenderink et A. van Doorn. *The internal representation of solid shape with respect to vision*. Biological Cybernetics, vol. 32, no. 1, pages 211–216, 1979. (Cité en page 10.)
- [Kolhatkar 2004] R. Kolhatkar. Grassmann varieties. Master's thesis, McGill University, 2004. (Cité en page 44.)
- [Kontkanen 2006] J. Kontkanen et T. Aila. *Ambient Occlusion for Animated Characters*. In Thomas Akenine-Möller Wolfgang Heidrich, editeur, Rendering Techniques 2006 (Eurographics Symposium on Rendering). Eurographics, jun 2006. (Cité en page 79.)
- [Kouyoumjian 1974] R. G. Kouyoumjian et P. H. Pathak. *A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface*. Proc. IEEE, vol. 62, no. 11, pages 1448–1461, November 1974. (Cité en page 97.)
- [Lafortune 1997] E. P. Lafortune, S.-C. Foo, K. E. Torrance et D. P. Greenberg. *Non-linear approximation of reflectance functions*. In SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 117–126, 1997. (Cité en pages 83 et 85.)
- [Laine 2005a] S. Laine et T. Aila. *A Hierarchical Penumbra Casting*. In Proceedings of Eurographics 2005, pages 313–322. Eurographics Association, 2005. (Cité en page 63.)
- [Laine 2005b] S. Laine, T. Aila, U. Assarsson, J. Lehtinen et T. Akenine-Möller. *Soft Shadow Volumes for Ray Tracing*. ACM Transactions on Graphics (SIGGRAPH 2005), vol. 24, no. 3, pages 1156–1165, 2005. (Cité en page 63.)
- [Laine 2010] S. Laine et T. Karras. *Two Methods for Fast Ray-Cast Ambient Occlusion*. Computer Graphics Forum, vol. 29, no. 4, pages 1325–1333, 2010. (Cité en page 79.)
- [Lambert 1760] Johann Heinrich Lambert. *Photometria, sive, de mensura et gradibus luminis, colorum et umbrae*. V.E. Klett, Augustae Vindelicorum :, 1760. (Cité en page 79.)
- [Landis 2002] Hayden Landis. *Production-Ready Global Illumination*. In Siggraph Course Notes, volume 16, 2002. (Cité en page 78.)
- [Lastra 2002] M. Lastra, C. Ureña, J. Revelles et R. Montes. *A Particle-Path based Method for Monte-Carlo Density Estimation*. In Eurographics Workshop on Rendering, 2002. (Cité en page 74.)
- [Lavignotte 2002] F. Lavignotte et M. Paulin. *A New Approach of Density Estimation for Global Illumination*. In WSCG, Plzen , Czech Republic, pages 263–273, Plzen , Czech Republic, February 2002. University of West Bohemia. (Cité en page 74.)

- [Lavignotte 2003] F. Lavignotte et M. Paulin. *Scalable photon splatting for global illumination*. In GRAPHITE '03, pages 203–ff, New York, NY, USA, 2003. ACM. (Cité en pages 74 et 165.)
- [Lazard 2010] S. Lazard, C. Weibel, S. Whitesides et L. Zhang. *On the computation of 3D visibility skeletons*. In Proceedings of the 16th annual international conference on Computing and combinatorics, pages 469–478, 2010. (Cité en page 11.)
- [Lemieux 2009] C. Lemieux. Monte carlo and quasi-monte carlo sampling. Springer Series in Statistics. Springer, New-York, 2009. (Cité en page 118.)
- [Liebe 2010] C. Liebe, P. Combeau, A. Gaugue, Y. Pousset, L. [Avenueau](#), R. Vauzelle et J.-M. Ogier. *Ultra-Wideband Indoor Channel Modelling Using Ray-Tracing Software for through-the-Wall Imaging Radar*. International Journal of Antennas and Propagation, vol. 2010, pages Article ID 934602, 14 pages, Juin 2010. (Cité en pages 110 et 112.)
- [Lienhardt 1989] Pascal Lienhardt. *Subdivisions of n-dimensional spaces and n-dimensional generalized maps*. In Proceedings of the fifth annual symposium on Computational geometry, SCG '89, pages 228–236, 1989. (Cité en page 88.)
- [Lloyd 2004] B. Lloyd, J. Wend, N. K. Govindaraju et D. Manocha. *CC Shadow Volumes*. In Rendering Techniques, pages 197–206, 2004. (Cité en page 63.)
- [Mahovsky 2006] J. Mahovsky et B. Wyvill. *Memory-Conserving Bounding Volume Hierarchies with Coherent Raytracing*. Computer Graphics Forum, vol. 25, no. 2, pages 173–182, 2006. (Cité en page 88.)
- [Marcus 1975] M. Marcus. Finite dimensional multilinear algebra. Marcel-Dekker, 1975. (Cité en page 44.)
- [Markovic 2010] D. Markovic, A. Canclini, F. Antonacci, A. Sarti et S. Tubaro. *Visibility-Based Beam Tracing for Soundfield Rendering*. In IEEE Intl. Workshop on Multimedia Signal Processing (MMSp-2010), pages 40–45, Oct. 2010. (Cité en page 17.)
- [Masson 2010] E. Masson, P. Combeau, Y. Cocheril, M. Berbineau, L. [Avenueau](#) et R. Vauzelle. *Radio wave propagation in arch-shaped tunnels : Measurements and simulations by asymptotic methods*. Comptes Rendus de l'Académie des Sciences - Series IV - Physics, vol. 11, no. 1, pages 44 – 53, Janvier 2010. (Cité en page 117.)
- [Masson 2011] E. Masson, Y. Cocheril, P. Combeau, L. [Avenueau](#), M. Berbineau, R. Vauzelle et E. Fayt. *Radio wave propagation in curved rectangular tunnels at 5.8 GHz for metro applications, simulations and measurements*. EURASIP Journal on Wireless Communications and Networking, vol. 2011, no. 1, page 202, 2011. (Cité en page 118.)
- [Masson 2013] E. Masson, P. Combeau, Y. Cocheril, L. [Avenueau](#), M. Berbineau et R. Vauzelle. *Ray Launching modeling in curved tunnels with rectangular or non rectangular section*. In Dr. Yi Zheng, editeur, Wave Propagation Theories and Applications. InTech, Février 2013. (Cité en page 118.)
- [McGuire 2010] Morgan McGuire. *Ambient Occlusion Volumes*. In Proceedings of High Performance Graphics 2010, June 2010. (Cité en pages 79 et 82.)

- [Méndez-Feliu 2009] A. Méndez-Feliu et M. Sbert. *From obscurances to ambient occlusion : A survey*. The Visual Computer, vol. 25, pages 181–196, 2009. (Cité en page 78.)
- [Méndez 2003] A. Méndez, M. Sbert et L. Neumann. *Obscurances for ray-tracing*. Eurographics 2003 Poster Presentation, 2003. (Cité en pages 78 et 79.)
- [Méndez 2004] A. Méndez et M. Sbert. *Comparing hemisphere sampling techniques for obscurances computation*. In Proceedings of the International Conference on Computer Graphics and Artificial Intelligence (3IA 2004), 2004. (Cité en page 78.)
- [Meneveaux 1998] D. Meneveaux, E. Maisel et K. Bouatouch. *A New Partitioning Method for Architectural environments*. Novembre 1998. Journal of Vizualisation and Computer Animation, Volume 9 (1998), Issue 4, Wiley Publishers, pp 195-213. (Cité en page 88.)
- [Meunier 2010] S. Meunier, R. Perrot, L. [Avenueau](#), D. Meneveaux et D. Ghazanfarpour. *Cosine Lobes for Interactive Direct Lighting in Dynamic Scenes*. Computers & Graphics, vol. 34, no. 6, pages 767–778, Décembre 2010. (Cité en pages 83 et 86.)
- [Mittring 2007] Martin Mittring. *Finding next gen : CryEngine 2*. In ACM SIGGRAPH 2007 courses, pages 97–121, New York, USA, 2007. ACM. (Cité en page 79.)
- [Mora 2005a] F. Mora et L. [Avenueau](#). *Fast and Exact Direct Illumination*. In Computer Graphics International 2005 Proceedings, volume 23, pages 191–197, Stony Brooks University, New York, USA, June 2005. Computer Graphics International, IEEE Computer Society. (Cité en pages 39 et 70.)
- [Mora 2005b] F. Mora et L. [Avenueau](#). *Optimised Scanning of a Visibility Graph Data Structure for Efficient Ray-Tracing*. In Proceedings of IEEE European Conference on Wireless Technology, Paris, Octobre 2005. (Cité en pages 39 et 109.)
- [Mora 2005c] F. Mora, L. [Avenueau](#) et M. Mériaux. *Coherent and exact polygon-to-polygon visibility*. In Vaclav Skala, editeur, WSCG'2005 Full Papers Conference Proceedings, volume 13, pages 87–94, University of West Bohemia, Plzen, Czech Republic, february 2005. Winter School on Computer Graphics, UNION Agency – Science Press. (Cité en page 39.)
- [Mora 2006] Frédéric Mora. *Visibilité polygone à polygone : calcul, représentation, applications*. PhD thesis, Université de Poitiers, juillet 2006. (Cité en pages 9, 12, 29, 33, 34, 37 et 39.)
- [Mora 2012] F. Mora, L. [Avenueau](#), O. L. Apostu et D. Ghazanfarpour. *Lazy visibility evaluation for exact soft shadows*. Computer Graphics Forum, vol. 31, no. 1, pages 132–145, Février 2012. (Cité en pages 40, 58, 61, 70, 72 et 75.)
- [Mora 2013] F. Mora, L. [Avenueau](#), O.L. Apostu et D. Ghazanfarpour. *Lazy visibility evaluation for exact soft shadows*. In Proceedings of the Eurographics Conference, Girona, Spain, Mai 2013. (Cité en pages 58 et 70.)
- [Naylor 1990a] B. Naylor, J. Amanatides et Thibault. *Merging BSP trees yields polyhedral set operations*. In Proceedings of the 17th annual conference on Computer graphics and interactive techniques, pages 115–254. ACM Press, 1990. (Cité en page 30.)

- [Naylor 1990b] Bruce Naylor. *Binary space partitioning trees as an alternative representation of polytopes*. *Comput. Aided Des.*, vol. 22, no. 4, pages 250–252, 1990. (Cité en page 28.)
- [Ng 2003] R. Ng, R. Ramamoorthi et P. Hanrahan. *All-frequency shadows using non-linear wavelet lighting approximation*. *ACM Transactions on Graphics (TOG)*, vol. 22, pages 376–381, 2003. (Cité en page 176.)
- [Nirenstein 2002] S. Nirenstein, E. Blake et J. Gain. *Exact from-region visibility culling*. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 191–202. Eurographics Association, June 2002. (Cité en pages 12, 28, 50 et 66.)
- [Nirenstein 2003] Shaun Nirenstein. *Fast and accurate visibility preprocessing*. PhD thesis, University of Cape Town, October 2003. (Cité en pages 2, 12, 27, 29, 30, 32, 33, 34 et 53.)
- [Nishita 1983] T. Nishita et E. Nakamae. *Half-Tone Representation of 3-D Objects Illuminated by Area Sources or Polyhedron Sources*. In *Proceedings of IEEE Computer Software and Application Conference*, pages 237–342, 1983. (Cité en page 65.)
- [Nishita 1985a] T. Nishita et E. Nakamae. *Continuous tone representation of three-dimensional objects taking account of shadows and interreflection*. In *SIGGRAPH*, pages 23–30. ACM, 1985. (Cité en pages 63 et 65.)
- [Nishita 1985b] T. Nishita, I. Okamura et E. Nakamae. *Shading models for point and linear sources*. *ACM Trans. Graph.*, vol. 4, no. 2, pages 124–146, 1985. (Cité en page 63.)
- [Parrochia 2012] D. Parrochia, A. Micali et P. Anglès. *L'unification des mathématiques*. Collection informatique. Hermès/Lavoisier, Juillet 2012. (Cité en page 40.)
- [Parzen 1962] Emanuel Parzen. *On Estimation of a Probability Density Function and Mode*. *The Annals of Mathematical Statistics*, vol. 33, no. 3, pages 1065–1076, 1962. (Cité en page 159.)
- [Pattanaik 1995] S. N. Pattanaik et S. P. Mudur. *Adjoint equations and random walks for illumination computation*. *ACM Trans. Graph.*, vol. 14, no. 1, pages 77–102, 1995. (Cité en page 120.)
- [Pellegrini 1990] Marco Pellegrini. *Stabbing and ray shooting in 3 dimensional space*. In *SCG'90 : Proceedings of the 6th annual symposium on Computational Geometry*, pages 177–186, New York, USA, 1990. ACM Press. (Cité en page 29.)
- [Pellegrini 1991] Marco Pellegrini. *Ray-shooting and isotopy classes of lines in 3-dimensional space*. In Frank Dehne, Jörg-Rüdiger Sack et Nicola Santoro, éditeurs, *Algorithms and Data Structures*, volume 519, pages 20–31. Springer, 1991. (Cité en pages 59 et 80.)
- [Pellegrini 1993] Marco Pellegrini. *Ray shooting on triangles in 3-space*. *Algorithmica*, vol. 9, no. 5, pages 471–494, 1993. (Cité en pages 12, 29, 61 et 80.)
- [Pellegrini 2004] Marco Pellegrini. *Ray Shooting and Lines in Space*. In Jacob E. Goodman et Joseph O'Rourke, éditeurs, *Handbook of discrete and computational geometry - second edition*, pages 839–856. Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2004. (Cité en page 80.)

- [Perrot 2012] Romuald Perrot. *Lobes de cosinus et visibilité pour la simulation d'éclairage*. PhD thesis, Université de Poitiers, Décembre 2012. (Cité en pages 135, 145, 157 et 169.)
- [Petitjean 1992] Sylvain Petitjean. Computing exact aspect graphs of smooth objects bounded by smooth algebraic surfaces. Master's thesis, University of Illinois, Urbana Champaign, IL, June 1992. (Cité en page 10.)
- [Pharr 2010] M. Pharr et G. Humphreys. *Physically Based Rendering, Second Edition : From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd édition, 2010. (Cité en page 163.)
- [Platinga 1990] W.H. Platinga et C. R. Dyer. *Visibility, occlusion and the aspect graph*. International Journal of Computer Vision, vol. 5, no. 2, pages 137–160, 1990. (Cité en page 10.)
- [Pocchiola 1993] Michel Pocchiola. *The visibility complex*. In Proceedings of the ninth annual symposium on Computational Geometry, pages 328–337. ACM Press, 1993. (Cité en page 10.)
- [Popov 2007] S. Popov, J. Günther, H.-P. Seidel et P. Slusallek. *Stackless KD-Tree Traversal for High Performance GPU Ray Tracing*. Computer Graphics Forum, vol. 26, no. 3, pages 415–424, 2007. (Cité en page 88.)
- [Pu 1998] Fan-Tao Pu. *Data structures for global illumination computation and visibility queries in 3-space*. PhD thesis, Chair-David M. Mount, 1998. (Cité en pages 12 et 28.)
- [Pu 1999] F.-T. Pu et D. M. Mount. *Binary Space Partitions in Plücker space*. In Selected papers from the International Workshop on Algorithm Engineering and Experimentation, pages 94–113, London, UK, 1999. ALENEX'99, Springer-Verlag. (Cité en pages 28 et 59.)
- [Purcell 2002] T. J. Purcell, I. Buck, W. R. Mark et P. Hanrahan. *Ray tracing on programmable graphics hardware*. ACM Trans. Graph., vol. 21, no. 3, pages 703–712, Juillet 2002. (Cité en page 88.)
- [Ramamoorthi 2007] R. Ramamoorthi, D. Mahajan et P. Belhumeur. *A first-order analysis of lighting, shading, and shadows*. ACM Trans. Graph., vol. 26, no. 1, Janvier 2007. (Cité en page 129.)
- [Reshetov 2005] A. Reshetov, A. Soupikov et J. Hurley. *Multi-level ray tracing algorithm*. ACM Trans. Graph., vol. 24, no. 3, pages 1176–1185, 2005. (Cité en page 88.)
- [Reveillès 1995] J. P. Reveillès. *Combinatorial Pieces In Digital Lines And Planes*. SPIE Vision Geometry IV, vol. 2573, 1995. (Cité en page 101.)
- [Reynaud 2005a] S. Reynaud, Y. Cocheril, R. Vauzelle, A. Reineix, L. [Avenueau](#), M. Khoudeir et C. Guiffaut. *Influence d'une Description Fine de l'Environnement sur le Canal de Propagation Indoor*. In JSCNFRS '05, Journées Scientifiques du Comité National Français de Radioélectricité Scientifique, Février 2005. (Cité en page 105.)

- [Reynaud 2005b] S. Reynaud, Y. Cocheril, R. Vauzelle, L. Aveneau et A. Reineix. *Influence of an Accurate Environment Description for the Indoor Propagation Channel Modelling*. In ECWT '05 Proceedings, European Conference on Wireless Technology, Octobre 2005. (Cité en page 105.)
- [Rubin 1980] S. M. Rubin et T. Whitted. *A 3-dimensional representation for fast rendering of complex scenes*. SIGGRAPH Comput. Graph., vol. 14, no. 3, pages 110–116, Juillet 1980. (Cité en page 88.)
- [Schjøth 2007] L. Schjøth, J. R. Frisvad, K. Erleben et J. Sporring. *Photon differentials*. In Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia, pages 179–186, New York, NY, USA, 2007. ACM. (Cité en page 164.)
- [Schregle 2003] Roland Schregle. *Bias Compensation for Photon Maps*. Computer Graphics Forum, vol. 22, pages 729–742, 2003. (Cité en pages 73, 74, 164 et 165.)
- [Schregle 2005] R. Schregle. *Daylight Simulation with Photon Maps*. Fraunhofer IRB Verlag, 2005. (Cité en page 165.)
- [Schubert 1879] Hermann Schubert. *Kalkul der abzählenden geometrie*. Springer Verlag, 1879. (Cité en page 11.)
- [Shinya 1987] M. Shinya, T. Takahashi et S. Naito. *Principles and applications of pencil tracing*. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques, SIGGRAPH '87, pages 45–54, 1987. (Cité en page 88.)
- [Silverman 1998] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1998. (Cité en pages 161 et 162.)
- [Simonoff 1996] J.S. Simonoff. *Smoothing methods in statistics*, 1996. (Cité en page 163.)
- [Sloan 2002] P.-P. Sloan, J. Kautz et J. Snyder. *Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments*. In ACM Transactions on Graphics, pages 527–536, 2002. (Cité en pages 83 et 176.)
- [Sloan 2003] P.-P. Sloan, J. Hall, J. Hart et J. Snyder. *Clustered principal components for precomputed radiance transfer*. ACM Transactions on Graphics (TOG), 2003. (Cité en page 176.)
- [Sottile 2002] F. Sottile et T. Theobald. *Lines tangent to $2n - 2$ spheres in \mathbb{R}^n* . Transaction of the American Mathematical Society, vol. 354, pages 4815–4829, 2002. (Cité en page 11.)
- [Sottile 2003] Frank Sottile. *Enumerative Real Algebraic Geometry*. In S. Basu et L. Gonzalez-Vega, éditeurs, *Algorithmic and Quantitative Aspects of Real Algebraic Geometry*, volume 60 of *AMS DIMACS series*, pages 139–180, 2003. (Cité en page 11.)
- [Sottile 2007a] F. Sottile, H. Brönnimann, O. Devillers et S. Lazard. *Lines tangent to four triangles in three-dimensional space*. Discrete and Computational Geometry, vol. 37, no. 3, pages 369–380, 2007. (Cité en page 11.)
- [Sottile 2007b] F. Sottile et T. Theobald. *Line problems in nonlinear computational geometry*. ArXiv Mathematics e-prints, April 2007. (Cité en page 11.)

- [Stolfi 1991] Jorge Stolfi. *Oriented projective geometry : A framework for geometric computations*. Academic Press Professional, Inc., San Diego, CA, USA, 1991. (Cité en pages 16 et 23.)
- [Szirmay-Kalos 2010] L. Szirmay-Kalos, T. Umenhoffer, B. Tóth, L. Szécsi et M. Sbert. *Volumetric Ambient Occlusion for Real-Time Rendering and Games*. IEEE Comput. Graph. Appl., vol. 30, no. 1, pages 70–79, 2010. (Cité en page 79.)
- [Tanaka 1997] T. Tanaka et T. Takahashi. *Fast Analytic Shading and Shadowing for Area Light Sources*. In *Computer Graphics Forum (Eurographics '97)*, volume 16(3), pages 231–240, September 1997. (Cité en page 63.)
- [Tarini 2006] M. Tarini, P. Cignoni et C. Montani. *Ambient occlusion and edge cueing to enhance real time molecular visualization*. IEEE Transaction on Visualization and Computer Graphics, vol. 12, no. 6, sep/oct 2006. (Cité en page 79.)
- [Teller 1991a] S. Teller et M. E. Hohmeyer. *Stabbing oriented convex polygons in randomized $O(n^2)$ time*. Rapport technique UCB/CSD/91/669, Computer Science Department, University of California, Berkeley, July, 1991. (Cité en pages 12 et 25.)
- [Teller 1991b] S. J. Teller et C. H. Séquin. *Visibility preprocessing for interactive walkthroughs*. SIGGRAPH Comput. Graph., vol. 25, no. 4, pages 61–70, Juillet 1991. (Cité en page 88.)
- [Teller 1994] S. Teller et P. Hanrahan. *Global Visibility Algorithms for Illumination Computations*. In *Proceeding of SIGGRAPH'94*, pages 443–450. ACM Press, 1994. (Cité en page 12.)
- [Teller 1999] S. Teller et M. Hohmeyer. *Determining the Lines Through Four Lines*. Journal of Graphics Tools, vol. 4, no. 3, pages 11–22, 1999. (Cité en page 50.)
- [Tenorio 2011] A. F. Tenorio et L. [Aveneau](#). *Some Considerations about Geometric Algebras in relation with Visibility in Computer Graphics*. Imagen-A, vol. 2, no. 1, pages 21–24, 2011. (Cité en page 128.)
- [Theobald 2002] Thorsten Theobald. *An enumerative geometry framework for algorithmic line problems in \mathbb{R}^3* . SIAM Journal on Computing, vol. 31, no. 4, pages 1212–1228, 2002. (Cité en page 11.)
- [Tobler 2006] R. F. Tobler et S. Maierhofer. *Improved Illumination Estimation for Photon Maps in Architectural Scenes*. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pages 257–262, 2006. (Cité en pages 73, 74, 75 et 166.)
- [Torkestani 2012] S.S. Torkestani, S. Sahuguede, A. Julien-Vergonjanne, L. [Aveneau](#), P. Combeau et O. Gueu. *Outage Probability Performance in Mobile Indoor Optical Wireless Communication Environment*. In *Proceedings of the IWOW*, Pise, Italie, Octobre 2012. 3 pages. (Cité en page 133.)
- [Torres 1999] R. P. Torres, L. Valle, M. Domingo, S. Loredó et M. C. Diez. *CINDOOR : An engineering tool for planning and design of wireless systems in enclosed spaces*. IEEE Antennas and Propagation Magazine, vol. 41, no. 4, 1999. (Cité en page 116.)

- [Veach 1997] Eric Veach. *Robust Monte-Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997. <http://graphics.stanford.edu/papers/veach{ }thesis>. (Cité en pages 120, 141 et 163.)
- [Wächter 2006] C. Wächter et A. Keller. *Instant ray tracing : the bounding interval hierarchy*. In Proceedings of the 17th Eurographics conference on Rendering Techniques, EGSR'06, pages 139–149, 2006. (Cité en pages 88, 100 et 131.)
- [Wald 2001] I. Wald, P. Slusallek, C. Benthin et M. Wagner. *Interactive Rendering with Coherent Ray Tracing*. vol. 20, no. 3, pages 153–164, 2001. (Proceedings of EUROGRAPHICS). (Cité en page 88.)
- [Wald 2004] Ingo Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Saarland University, 2004. (Cité en page 65.)
- [Wald 2006] I. Wald, T. Ize, A. Kensler, A. Knoll et S. G. Parker. *Ray Tracing Animated Scenes using Coherent Grid Traversal*. In In Proceedings of ACM SIGGRAPH, 2006. (Cité en page 88.)
- [Wald 2007] I. Wald, S. Boulos et P. Shirley. *Ray tracing deformable scenes using dynamic bounding volume hierarchies*. ACM Trans. Graph., vol. 26, no. 1, Janvier 2007. (Cité en page 88.)
- [Wang 2006] T. S. Wang et C. F. Yang. *Simulations and Measurements of Wave Propagations in Curved Road Tunnels for Signals From GSM Base Stations*. IEEE Transactions on Antennas and Propagation, vol. 54, no. 9, pages 2577–2584, 2006. (Cité en pages 116 et 117.)
- [Whitted 1980] Turner Whitted. *An Improved Illumination Model for Shaded Display*. In Communications of the ACM, pages 343–349. Communications of the ACM, Juin 1980. (Cité en page 88.)
- [Zamora-Esquivel] Julio Zamora-Esquivel. *$G(6, 3)$ Geometric Algebra*. In 9th International Conference on Clifford Algebras and their Applications in Mathematical Physics. (Cité en page 128.)
- [Zhang 2007] L. Zhang, H. Everett, S. Lazard et S. Whitesides. *Towards an implementation of the 3D visibility skeleton*. In Proceedings of the twenty-third annual symposium on Computational geometry, SCG '07, pages 131–132, New York, USA, 2007. ACM. (Cité en page 11.)
- [Zhukov 1998] S. Zhukov, A. Iones et G. Kronin. *An Ambient Light Illumination Model*. In Eurographics Symposium on Rendering, pages 45–56, 1998. (Cité en pages 78 et 79.)

Résumé : Ce mémoire d'habilitation à diriger des recherches résume les différents travaux menés entre 2000 et 2013 au sein du laboratoire SIC, à l'Université de Poitiers. Le coeur de ces activités est la caractérisation, le calcul, la représentation et l'utilisation de la visibilité entre des ensembles convexes de points de l'espace géométrique en dimension $n \geq 2$. Nous avons notamment utilisé l'algèbre de Grassmann pour construire un espace de droites, dans lequel le problème de la visibilité peut être discuté. Nous avons proposé deux méthodes de calculs, l'un explicite et complet, et l'autre reposant sur une évaluation paresseuse. Différentes applications en synthèse d'images reposent sur ces travaux. D'autres travaux autour de la visibilité, en synthèse d'images mais aussi dans le domaine de la propagation des ondes, sont présentés dans ce mémoire. L'ensemble a été réalisé à travers l'encadrement de 5 doctorants.

Mots clés : Visibilité, Rendu, Ombres Douces, Simulation Radio, Simulation Optique

Abstract : This document presents the research work we made between 2000 and 2013 at the SIC Laboratory, University of Poitiers. The heart of these activities is the characterization, the computation, the representation and the usage of the visibility between convex sets of points in a geometrical space of dimension $n \geq 2$. We have used the Grassmann algebra to define a set of lines, allowing us to prove many new results. We have proposed two new computation techniques : the first is an explicit and produce a full visibility information, while the second one is a lazy evaluation method. Many works have been made using these algorithms, from soft shadows rendering in rendering to wave propagation simulation. These works have been done through the supervision of five PhD students.

Keywords : Visibility, Rendering, Soft Shadows, Radio Simulation, Optical Simulation
